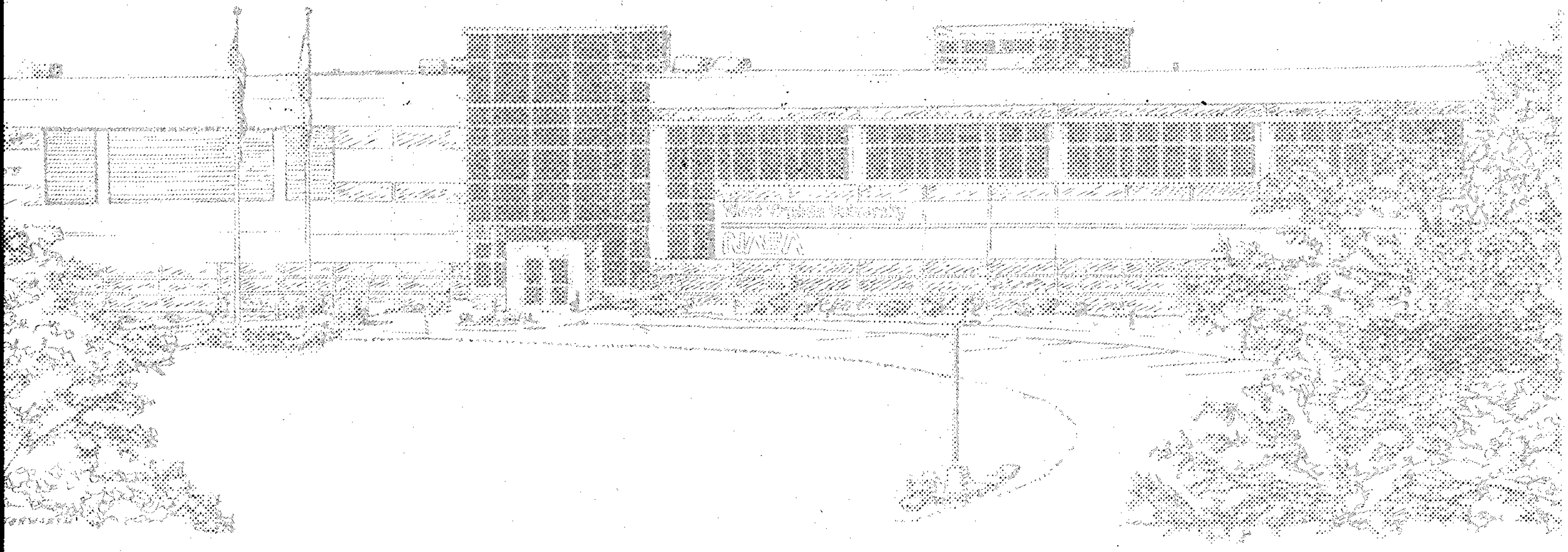
NASA-IVV-96-019
WVU-SRL-96-019
WVU-SCS-TR-96-30

IN-61-CR
008201

NASA/WVU Software IV & V Facility
Software Research Laboratory
Technical Report Series

The Application of V&V Within Reuse-Based Software Engineering

by Edward Addy



National Aeronautics and Space Administration

West Virginia University

The Application of V&V Within Reuse-Based Software Engineering

Edward Addy

NASA/WVU Software Research Laboratory
100 University Drive, Fairmont, WV 26554
Tel: (304) 367-8353
Fax: (304) 367-8211
Email: eaddy@wvu.edu

Abstract

Verification and Validation (V&V) is performed during application development for many systems, especially safety-critical and mission-critical systems. The V&V process is intended to discover errors as early as possible during the development process. Early discovery is important in order to minimize the cost and other impacts of correcting these errors.

In reuse-based software engineering, decisions on the requirements, design and even implementation of domain assets can be made prior to beginning development of a specific system. In order to bring the effectiveness of V&V to bear within reuse-based software engineering, V&V must be incorporated within the domain engineering process.

Keywords: Reuse, verification, validation, domain engineering, architecture, V&V

Workshop Goals: Learn; network; receive feedback and comments on research.

Working Groups:reuse and product lines; component certification; rigorous behavioral specification; domain engineering helps manage change.

According to the terms of Cooperative Agreement #NCCW-0040, the following approval is granted for distribution of this technical report outside the NASA/WVU Software Research Laboratory

George J Sabolish

George J. Sabolish Date
Manager, Software Engineering

John R. Callahan Date
WVU Principal Investigator

According to the terms of Cooperative Agreement #NCCW-0040, the following approval is granted for distribution of this technical report outside the NASA/WVU Software Research Laboratory

George J Sabolish

George J. Sabolish Date
Manager, Software Engineering

John R. Callahan Date
WVU Principal Investigator

1 Background

Mr. Addy performs research in the area of software verification and validation (V&V), and he has a special interest in the application of V&V within reuse-based software engineering. He led a working group at Reuse '96 that discussed a framework proposed by Mr. Addy for performing V&V within reuse-based software engineering.

Mr. Addy was the lead author for the draft Department of the Navy Reuse Implementation Plan and Implementation Guide. He was a member of a small team that investigated methodologies for performing domain analysis and performed a prototype domain analysis for the Program Executive Office of Cruise Missiles and Unmanned Aerial Vehicles. He is Recorder for the Reuse Issues Action Team, a subgroup of the ACM SIGAda Reuse Working Group that addresses management and business issues related to software reuse.

2 Position

2.1 Verification and Validation in Traditional System Application Engineering

V&V is performed during application system development, within the context of many different development methodologies. V&V is a set of activities performed in parallel with system development and designed to provide assurance that a software system meets the operational needs of the user[Lew92]. It ensures that the requirements for the system are correct, complete, and consistent, and that the life-cycle products correctly implement system requirements. The V&V process evaluates software in a systems context, using a structured approach to analyze and test the software against system functions and against hardware, user and other software interfaces[WF89b].

A set of V&V activities is defined in the IEEE Standard for Software Verification and Validation Plans[Soc86]. These activities are divided into the life-cycle phases listed below.

- Management of V&V
- Concept Phase V&V
- Requirements Phase V&V
- Design Phase V&V
- Implementation Phase V&V
- Test Phase V&V
- Installation and Checkout Phase V&V
- Operations and Maintenance Phase V&V

V&V is structured to detect high-risk errors early during the application development process[WF89a]. The earlier a problem is discovered in the development process, the less costly it is to correct the problem. To take advantage of this, V&V begins verification within system application development at the concept or high-level requirements phase. However, a reuse-based software development process has tasks that are performed earlier, and possibly much earlier, than high-level requirements for a particular application system. In order to bring the effectiveness of V&V to bear within a reuse-based software engineering process, V&V must be incorporated within the domain engineering process.

2.2 Verification and Validation in Reuse-Based Software Engineering

One model for reuse-based software engineering is the STARS Two Life-Cycle Model[fARSS96]. This model divides product line development into the two primary activities of domain engineering and application engineering. The primary domain engineering tasks are domain analysis, domain design, and domain implementation, while the primary application engineering tasks are requirements analysis, system design, and system implementation. The model assumes a domain-specific, architecture-centered approach to software reuse.

Edward Addy created a framework for performing V&V within reuse-based software engineering by adding V&V activities to the STARS Two Life-Cycle Model. A working group at Reuse '96 revised the framework, and considered how the new domain-level and transition-level tasks would impact the scope and level of the traditional application-level tasks[Add96]. The resultant product is a model for performing V&V within reuse-based software engineering.

Application-level V&V tasks ensure the application products fulfill the requirements established during previous life-cycle phases. These tasks are those that are done in traditional system-oriented V&V. Domain-level V&V tasks are performed to ensure that domain products fulfill the requirements established during earlier phases of domain engineering. The domain-level tasks are similar to the traditional tasks, but are performed on the domain products rather than the system products. Transition-level tasks provide assurance that an application artifact correctly implements the corresponding domain artifact. The transition-level tasks are a new type of V&V task.

The V&V within reuse-based software engineering model uses the term "correspondence analysis" to describe the activities to provide assurance that an application artifact is a correct implementation of the domain artifact. Four activities are to be performed during correspondence analysis:

- Map the application artifact to the corresponding domain artifact.
- Ensure that the application artifact has not been modified from the domain artifact without proper documentation.
- Ensure that the application artifact is a correct instantiation of the domain artifact.
- Obtain information on testing and analysis on a domain artifact to aid in V&V planning for the application artifact.

The IEEE Standard for Software Verification and Validation Plans lists more phases for V&V than there are phases within the two life-cycle models. The following table shows the mapping of the IEEE Standard phases to the two life-cycle model phases for both application engineering and domain engineering.

Domain maintenance and evolution should be handled in a manner similar to that described in the operations and maintenance phase of application-level V&V. Changes proposed to domain artifacts should be assessed by V&V to determine the impact of the proposed correction or enhancement. If the assessment determines that the change will impact a critical area or function within the

IEEE Standard 1012 Phase	*Two Life-Cycle Model Phase*	
	Application-Level	*Domain-Level*
Management	all phases	all phases
Concept	Requirements	Domain Analysis
Requirements	Requirements	Domain Analysis
Design	Design	Design
Implementation	Implementation	Implementation
Test	Implementation	Implementation
Installation and Checkout	Implementation	Implementation
Operations and Maintenance	as needed	as needed

Table 1: Mapping of IEEE Standard 1012 Phases to Two Life-Cycle Model Phases

domain, appropriate V&V activities should be repeated to assure the correct implementation of the change.

No application-level V&V tasks should be eliminated due to tasks being performed at the domain or transition levels. It might be possible to reduce the level of effort for some application-level tasks, in a case where the application artifact is used in an unmodified form from the domain component, or where the application artifact is an instantiation of the domain component through parameter resolution.

Communication of the V&V work products and results is vital to avoid the repetition of V&V tasks and to ensure that potential reusers could properly assess the status of reusable components. V&V work products and results should be associated with the component and made available to domain and application engineers. In some cases, the V&V might be directed at a grouping of components rather than at an individual component, and this information should also be available. The information that should be communicated should include the following:

- V&V Planning Decisions and Rationale
- V&V Analysis Activities
- V&V Test Cases and Procedures
- V&V Results and Findings

3 Comparison

Several groups are working in the area of component verification, as evidenced by working groups at each WISR since 1992 that dealt specifically or tangentially with component certification. The major reuse libraries have established policies and procedures for performing component certification or evaluation [IBM92] [SPS95]. Several groups are conducting research component-level certification [DK92][SE96]. The purpose of these certification or evaluation tasks is to provide assurance that the component meets some standard of quality and to inform potential users of the results of the evaluation. The evaluation is performed using the specification for the component or using constraints identified by the evaluator.

The work that is closest to the concepts described in the Problem section of this paper is the Product Line Asset Support (PLAS) program of the Electronic Systems Center (ESC), Air Force Material Command, USAF[SE96]. The ESC is working with an industry group to develop an architecture for several different product lines. The PLAS group will perform "suitability" testing to determine if a Commercial off-the-shelf (COTS) or Government off-the-shelf (GOTS) software product meets the architectural and functional requirements of a component area within a software architecture. The products will be tested using a standard process to provide an objective analysis of the functionality and architectural capabilities using criteria which is derived from the architecture.

In most of the work described above, the software components being evaluated are code components, either source code or executable code. In all of the work described above, the components are evaluated in isolation of other components. V&V within reuse-based software engineering proposes that the components be evaluated relative to the context in which they will be used. The domain engineering products provide the framework in which to evaluate subsequent domain components and application components. This process allows for early evaluation of all domain components, not simply code components. The process can be tailored to focus on critical areas of the domain, just as traditional V&V is tailored to focus on critical areas of the system.

References

[Add96] Edward A. Addy. V&V Within Reuse-Based Software Engineering. In *Proceedings for the Fifth Annual Workshop on Software Reuse Education and Training, Reuse '96*, number http://www.asset.com/WSRD/conferences/proceedings/results/addy/addy.html, August 1996.

[DK92] Michael F. Dunn and John C. Knight. Certification of Reusable Software Parts. Technical report, University of Virginia, August 1992.

[fARSS96] Software Technology for Adaptable Reliable Systems (STARS). DARPA STARS Overview. Technical Report http://www.asset.com/stars/darpa/Overview/ov8-plprc.html, Defense Advanced Research Projects Agency (DARPA), September 1996.

[IBM92] Corp. IBM. Evaluating Reusable Software Assets: Criteria and Procedures. Technical Report ASSET_A_252, ASSET, 1992.

[Lew92] Robert O. Lewis. *Verification and Validation, A Life Cycle Engineering Process for Quality Software.* John Wiley & Sons, 1992.

[SE96] Loral Defense Systems-East. Product Line Asset Support Concept of Operations. Technical Report STARS-VC-K017R1/001/00, Electronic Systems Command, 1996.

[Soc86] IEEE Computer Society. *IEEE Standard for Software Verification and Validation Plans.* Number IEEE Std 1012-1986. IEEE Computer Society, 1986.

[SPS95] Inc. Software Productivity Solutions. Operational Concept Document for the Automated Certification Environment (ACE), volume 1. Technical report, Rome Laboratory, September 1995.

[WF89a] Dolores R. Wallace and Roger U. Fujii. Software Verification and Validation: An Overview. *IEEE Software*, pages 10–17, May 1989.

[WF89b] Dolores R. Wallace and Roger U. Fujii. Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards. Technical Report NIST Special Publication 500-165, National Institute of Standards and Technology, 1989.

4 Biography

Edward A. Addy is a Research Associate for West Virginia University, working at the NASA/WVU Software Research Laboratory at the NASA/WVU Software IV&V Facility in Fairmont, WV. He performs research on the process, methods, tools of independent verification and validation, especially as related to software reuse and to safety-critical software. He works with NASA IV&V practitioners located at the IV&V Facility, and with NASA projects at other centers and labs.

Prior to joining the NASA/WVU Software Research Laboratory, Mr. Addy served as a senior computer scientist with Logicon, where he worked in the areas of safety analysis for cruise missile weapon control systems and software reuse for the Department of the Navy and for the Cruise Missile and Unmanned Aerial Vehicle program executive office. Mr. Addy serves as Recorder for the Reuse Issues Action Team, a subgroup to the ACM SIGAda Reuse Working Group. He is a member of ACM and of the IEEE Computer Society. He is in the graduate program in Computer Science at West Virginia University. He received an M.S. in Mathematics from Wake Forest University in 1979, and a B.S. in Mathematics (Education) from Michigan State University in 1975.