

The Role of Verification and Validation in System Development Life Cycle

Priyanka Upadhyay

(School of Computers & Electronics, IPS Academy, India)

Abstract: *Software is build to satisfy needs of specific client or business. In today's competitive world of software, software developers have to deliver quality product on time. Quality of software is very essential in software development and for this developer has to verify that the software is functioning properly and validate the product as per client requirement. Importance of Verification and Validation in software development is to maintain software quality. Verification and Validation techniques are applied to all phases of System Development Life Cycle. This paper presents verification and validation activities in the software development process, how quality of software can be improve, verification and validation objective and lifecycle usage.*

Keyword-*System Development Life Cycle, Software Quality, Validation, Verification.*

I. Introduction

In the age of the Internet, where software is more mission-critical than ever, it's no longer enough for your development projects to succeed some of the time. You need to deliver excellent software quality, consistently and do it faster than ever. Software Quality is achieved through the application of development techniques and the use of Verification and Validation (V&V) procedures through the development process. The goal of V&V of software is that we need to check if developed software meets the client's need & specification. V&V is a collection of analysis and testing activities across full life cycle and complements the efforts of other quality-engineering functions.

V&V comprehensively analyzes and tests software to determine that it performs its intended functions correctly, to ensure that it performs no unintended functions, and to measure its quality and reliability. V&V is a system-engineering discipline to evaluate software in a system context. Like systems engineering, it uses structured approach to analyzes and test the software against all system functions and against hardware, user and other software interfaces [1]. V&V are two branches of software testing. They are complementary to each other and not substitutes of each other. Proper V&V together can ensure the quality of software as well as quality of processes used for developing and testing software which will finally help in achieving business objectives. Only proper verification without validation and vice versa cannot ensure a good software product. Rather, it needs combination of both.

II. Objective Of V&V

Verification – Building the right software? [2].

Verification is concerned about the correctness of process. Verification determines whether the software of a given phase system development life cycle satisfies the requirements established during previous phase or not. Verification approaches attempt to identify product faults or errors, which gives rise to failure. For example building a weight machine for displays the weight only upon inserting one rupee coin. Now this requirement of the customer and verification determines whether the software satisfies the requirement or not.

Validation – Building the software right? [2].

Validation is concerned about the correctness of product. Validation is the process of evaluating the software at the end of its development to ensure that it is free from failures and complies with its requirements. To ensure that the product actually meets the client needs and the specifications are correct. For example once this weight machine is built it's the Tester who validate the weight machine for display the weight only for 1Rs coin according to user requirement. But if machine displays the weight for "2" and "5" rupee coin also then it's a bug in software and it need to be reported. Therefore, validation ensures that software works properly for correct inputs and gives error message for wrong inputs.

III. Verification And Validation In Life Cycle

This section presents V&V in software development life cycle. That is, what is checked in each of the life cycle phases and who perform this checking. Moreover, what are the techniques used to perform the checking. Software V&V employ review, analysis and testing techniques to determine whether a software system and its intermediate products comply with requirements. These requirements include both functional capabilities and quality attributes. V&V's primary task is to manage project risk by identifying and monitoring

errors throughout the development and maintenance process. Since it is impossible to identify and resolve all errors early in a project's life cycle, it is V&V contractor's task to identify the error as early as possible and track the progress towards their resolution.

Verification ensures that the system (software, hardware, documentation, and personnel) complies with an organization's standards and processes, relying on review or non-executable methods. Validation physically ensures that the system operates according to plan by executing the system functions through a series of tests that can be observed and evaluated.

Life Cycle Stages	Verification Activities
Requirement	Determine verification approach. Determine adequacy of requirements. Generate functional test data. Determine consistency of design with requirements.
Design	Determine adequacy of design. Generate structural and functional test data. Determine consistency with design.
Construction	Determine adequacy of implementation. Generate structural and functional test data for program. Apply test data.
Operation & Maintenance	Re-verify, commensurate with the level of redevelopment.

Figure1-Life cycle verification activities [3]

Verification requires several types of reviews, including requirements reviews, design reviews, code walk through, code inspections, and test reviews. The system user should be involved in these reviews to find defects before they are built into the system. In the case of purchased systems, user input is needed to assure that the supplier makes the appropriate tests to eliminate defects.

IV. Methods Of Verification

3.1.1. Self Review

One must capture the self review records and defects found in self review to improve the process. It is basically a self-learning or retrospection process. Self review is highly flexible with respect to time and defect finding, as one need not take an appointment for doing it. Defect found in self review can help in self improvement. Understanding-related defects may not be found in self-rev

3.1.2. Peer Review

A peer may be a fellow developer or tester as the case may be. There is also a possibility of superior review where peer is a supervisor with better knowledge and experience. Peer reviews are conducted frequently in SDLC at various stages of development. There are two types of review

3.1.2.1 Online Review

Author and reviewer meet together and review the work product jointly.

3.1.2.2 Offline Review

Author informs reviewer that product is ready and reviewer may review product as per his time availability.

3.1.3 Walkthrough

Walkthrough is a semi formal type of review as involves larger team along with the author reviewing a work product. It may involve a project team or a part of a project team doing a review jointly as the case may be. Each and every member present must be involved in making decisions. Walkthrough can be effectively used for communication between team.

V. Inspection (Formal Review)

It is a formal review where external people involved as 'inspector'. They are 'subject matter expert' who review the work product. Defects are recorded but solutions are not given by 'subject matter experts'. This helps the organization to initiate own action plan for fixing the defects.

3.1.5 Audits

Audit is a formal review based on samples. Audits are conducted by auditors who may or may not be experts in a given work product.

Software Validation activities may occur both during, as well as at the end of the software development life cycle to ensure that all requirements have been fulfilled.

VI. Methods Of Validation

Validation Example	Performed By	Explanation
Unit Testing	Developers	The testing of a single program, module or unit of code. Usually performed by the developer of the software performs as designed.
Integrated testing	Developers with support from an independent test team	The testing of related program, modules, or units of code. Validates that multiple parts of the system interact according to the system design.
System Testing	Independent Test Team	The testing of an entire computer system. This kind of testing can include functional and structural testing, such as stress testing. Validate the system requirements.
User Acceptance Testing	User with support independent test team	The testing of computer system or parts of computer system to make sure it will work in the system regardless of what the system requirements indicate.

VII. Role Of Verification And Validation

Role of V&V for each product must be determined by a project-by-project basis. This determination will be influenced by the criticality of the product, its constraints, and its complexity. In general, the objective of the V&V function is to insure that the product satisfies the user needs. Thus, everything in the product's requirements and specifications must be the target of some V&V activity. In order to limit the scope of this module, however, the V&V approaches described will concentrate on the functional and performance portions of the requirements and specifications with respect to safety, portability, usability, maintainability, serviceability, security, etc., although very important for many systems, will not be addressed here[4].

VIII. Improvement In Software Quality

Conformance to explicitly stated and agreed functional and non functional requirements may be termed as quality for the software product offered to customer/final users from their prospective [5]. Software quality interacts with each phase of every software development process. Planning should occur in the initial phase of a software project and should address the methods and techniques to be used in each phase. A description of each product should be defined in order to provide a basis for objectively identifying satisfactory completion of the phase [6].

The software quality management processes must address how well software products will, or do, satisfy customer requirements, provide value to the customers, and provide the software quality needed to meet software requirements. Some of the specific Software Quality Management processes are defined in standard (IEEE12207.0-96):

- Quality assurance process
- Verification process
- Validation process
- Review process
- Audit process

Software Quality Assurance(SQA) processes provide assurance that the software products and processes in the software life cycle conform to their specified requirements by planning, enacting, and performing a set of activities to provide adequate confidence that quality is being built into the software. SQA seeks to maintain the quality throughout the development and maintenance of the product by the execution of a variety of activities at each stage which can result in early identification of problems, an almost inevitable feature of any complex activity [7].

V&V addresses software product quality directly and uses testing techniques which can locate defects so that they can be addressed. It also assesses the intermediate products, however, and, in this capacity, the intermediate steps of the software life cycle processes. The V&V process determines whether or not products of a given development or maintenance activity conform to the requirement of that activity, and whether or not the final software product fulfills its intended purpose and meets user requirements. Verification is an attempt to ensure that the product is built correctly, in the sense that the output products of an activity meet the specifications imposed on them in previous activities. Validation is an attempt to ensure that the right product is built, that is, the product fulfills its specific intended purpose. V&V is a disciplined approach to assessing software products throughout the software development life cycle. A V&V effort strives to ensure that quality is built into the software and that the software satisfies user requirements.

IX. Conclusion And Perspective

This paper has argued that V&V are essential part of system development life cycle, because they offer the only way to judge the quality and success of software. V&V makes it sure that, certain rules are followed at the time of development of a software product and also makes it sure that the product that is developed fulfills the required specifications. This reduces the risk associated with any software project up to a certain level by helping in detection and correction of errors and mistakes, which are unknowingly done during the development process. Through literature survey of many literature sources as the methods used in this paper, we know that it is important for the V&V to be flexible due to changes of the software requirements during the software development. Therefore in future, we are going to focus on the V&V techniques that will help the software process improvement when there is change of functionality of the requirements of the software system.

References

- [1] Dolores R. Wallace, Roger U. Fujii, Software verification and validation an overview, Journal in IEEE Software Volume 6 Issue 3 May 1989 10-17.
- [2] Barry W.Boehm,Guidelines for verifying and validating software requirements and design specifications Euro IFIP 79, North Holland 1979,711-719.
- [3] W.Richard Adrion ,Martha A.Branstad and John.C.Chernavsky , Verification, Validation and testing of computer software, U.S.Dept of Commerce,National Bureau of Standards (Washington,D.C),1981.
- [4] James S.Collofello, Introduction to software verification and validation, SEI Curriculum Module SEI-CM-13-1.1 December 1988.
- [5] M.G. Limaye ,Software testing principal, techniques and tools(Tata McGraw-Hill Education, 2009)
- [6] Bradley J. Brown, Assurance of software quality, SEI Curriculum Module July 1987.
- [7] "Guide to the Software Engineering Body of Knowledge (SWEBOK)"
<http://www.computer.org/portal/web/swebok/html/ch11>