

The Role of Verification and Validation in System Development Life Cycle

Pranjali Kubde¹, Dhananjay Sable²
Computer Science and Engineering
Email: Pranjali.kubde@gmail.com¹

Abstract- The verification and validation are used to evaluate the software process. In the software development process, the developer verifies and validates the product as per customer's requirement. The V&V techniques are used various tools and approaches in software development life cycle like technical reviews, software testing, program verification etc. The verification and validation techniques are used in all phases in life cycle to improve and maintain the quality of software. V & V are the quality assurance activities which are used in life cycle. This paper presents the software verification and validation techniques in software development life cycle.

Index Terms- Software development life cycle, verification, Validation, Software quality.

1. Introduction

The verification and validation techniques are used in software development process to achieve software quality. Software engineering requires that software is verified during every phase of its development life cycle and validate when it is transferred. These activities are called 'Software verification and validation'.

The software verification and validation activities check the software against its specification. Every project must verify and validate the software it produces. The V&V is a disciplined approach to assessing software products throughout the product life cycle. It is a whole life cycle process that V & V must be applied at each stage in the software process. Verification and validation should establish confidence that the software is fit for purpose.

2. Objectives

Verification: "Are we building the product right?" The software should confirm to its specification. It is the correctness, completeness, consistency and accuracy of process. The verification refers to the process of determining the process of building the product is right or not.

Validation: "Are we building the right product?" The software should do what the user really requires. The validation is process of evaluating software at the end of the software development process to ensure compliance with software requirements.

3. Literature review

As part of this master's thesis, two large studies were done to support the framework design. Firstly, the state-of-the-art of the early V&V was studied and its results are presented in this section. Secondly, a survey on state-of-the-practice was carried out, which results are briefly analyzed in the

next sub-section. The state-of-the-art (SotA) study was extensive. A total of six researchers contributed to the study; the author of this thesis was the main author. The SotA report consisted of over 50 pages, with references to over 80 different sources. The SotA study was conducted in a structural way. First it defined the product lifecycle model and its phases. Then each phase of the life cycle was studied separately and the processes, methods and techniques for improving early V&V during these phases were identified. Then there was a study of the tools which are commonly used when doing V&V in general. Some features of the tools were listed and the tools were sorted based on their potential usage. The following sub-sections briefly present the study results.

3.2.1 V&V lifecycle model

In this section, the concept of software development lifecycle as well as the generic lifecycle model used for categorizing V&V methods is presented. This lifecycle model serves as the basis for building the framework. The contents of this section are adapted from the SotA study.

The product lifecycle begins with a definition that describes the expected behavior of the system. The solution is then designed, built and tested to compare behavior with its initial description. If it passes, it then gets deployed; if it doesn't then it gets reworked until it does pass. Lifecycle artifacts are the by-product of this evolution? These artifacts represent the system at particular stages of its life.

Requirements represent the behavior, designs represent the solution, and source code represents the implementation, and tests represent the qualifying argument for deployment.

V&V is a process in itself and it has a lifecycle of its own. The V&V lifecycle is run in parallel with its development. For example, as the behavior is defined and its by-product generated (e.g. requirements Specification), V&V will perform the

requirements analysis. Based on their assessment, the V&V process will gain an understanding of the system's behavior.

According to van Moll et al, the development of a complex product can be considered an ordered Sequence of phases and activities, regardless of the specific development approach (e.g. incremental, waterfall or evolutionary). In theory, each phase and activity is open to the injection of defects in the resulting information or work product due to many causes. The injected defects add to those that have been already injected in the previous phases and could propagate in the subsequent Phases. To be able to get the system "right the first time", the number of injected defects should be minimized, which is the task of defect prevention measures? In reality, defects will occur. Therefore, for any given phase, defect detection measures should be taken. These measures must reflect the typical type of defects injected and the information or work product produced. The goal is to minimize the amount of a defect that propagates in the subsequent phases. Ultimately, the number of residual defects in the end product should be as low as possible.

3.2.2 Techniques, methods and processes

The previous section presented five phases of a V&V lifecycle. Four of these phases were used in the State-of-the art study to introduce currently available methods, techniques and processes intended for early V&V. The concept V&V and the requirement V&V phases were combined in the study, and this combination will be called the requirements definition phase. In this sub-section, the techniques and methods identified in the study are listed. Each of these items is presented separately in the SotA document. The items were selected so that the basic V&V activities (see the previous chapter) would also be covered as thoroughly as possible. In the study, the methods and techniques were categorized based on the lifecycle phase so that the best ones (some of which could be applied also during the other phases of the development) could be applied.

4. Verification and validation in life cycle

In the software development life cycle, verification is human based checking of documents and files and this is static method for verifying design code. Validation always involves with executing the code. This is dynamic process for checking and testing the real product.

Life cycle stages	Verification Activities
Requirement	Analyzes end user information, system needs. Develops user requirements.
Design	Describe the features and

	operation like screen layouts, process diagrams, pseudo code etc.
Construction	The real code is written here. It is longest phase of software development life cycle.
Operation and maintenance	This process where care is taken for the developed product is known as maintenance

Figure 1: Life cycle verification activities.

5. Methods of verification

5.1 Review

A review is an overview of the whole process which includes the guidelines and specification.

5.2 Walkthrough

Walkthrough are the most time consuming and most formal of the informal method. Walkthrough is used for communication between team. For Ex. Software development team reviewing a product before the final product is sent for approval by the customer.

5.3 Inspection

Inspection is verification method that is used to compare how true the conceptual model matches the executable model.

5.4 Audit

An audit is used to establish how well a model matches the guidelines that are set in place.

6. Methods of validation

6.1 Unit Testing

Testing of a unit or module comparing it with requirements and make ready for integration. It is usually performed by the developer.

6.2 Integrated Testing

It is the systematic combination and testing of software components to insure consistency of component interfaces. It is usually performed by the developer as well as an independent test team.

6.3 System Testing

It is an integrated software system comparing it with software system requirements. It includes functional and structural testing. It is usually performed by the independent test team.

6.4 User Acceptance Testing

Testing is an integrated hardware and software system. It is performed by the user with support of independent test team.

7. Role of verification and validation

Verification tests are aimed at “building the system right”, and validation tests are aimed at “building the right system.” Thus, verification examines issues such as ensuring that the knowledge in the system is represented correctly, while validation examines procedures to ensure the system makes correct decisions.

Verifying and validating play critical roles in the development and implementation of case based systems. A priori, if the system is not verified then there may be errors in the case representations. If the system is not validated, then it may not make the desired quality of decisions.

8. Improvement of software quality

Some of the specific quality management processes are defined as follows.

8.1 Quality assurance process

Quality assurance (QA) is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions to customers. QA is applied to physical products in pre-production to verify what will be made meets specification and requirements, and during manufacturing production runs by validating lot samples meet specified quality controls. QA is also applied to software to verify that features and functionality meet business objective.

8.2 Validation Process

The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers.

8.3 verification process

The evolution of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process.

8.4 Review process

The review process provides a framework for reviewing assets where you can approve, retire, or delete assets. Users can review and verify the quality and accuracy of assets according to their specific area of expertise.

8.5 Audit Process

A software audit review, or software audit, is a type of software review in which one or more auditors who are not members of the software development organization conduct “An independent examination of a software product, software process, or set of software processes to assess compliance with specifications, standards, contractual agreements, or other criteria.”

Conclusion

We have surveyed many of the techniques used to validate the software systems. Of the methods discussed, the most successful techniques, such as walk-throughs, reviews, and inspection, applied to all stages in the life cycle. Discovery of errors within the first stage of development (requirements and design) is particularly critical since the cost of these errors escalates significantly if they remain undiscovered until construction or later. Until the development products at the requirements and design stages become formalized and hence to automated analysis, disciplined manual techniques will continue to be the key verification techniques.

REFERENCES

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