The Role of Verification and Validation in System Development Life Cycle

Mr. Rakesh Kumar, Mr. Narinder Kumar
S.B.S.B.M. University College Sardulgarh
Punjabi University Patiala

Abstract
In the age of the Internet, to make the software is more critical some time it is difficult to succeed in the development of project. It is necessary to deliver best software quality. It can be achieved through the use of Verification and Validation (V&V) procedures through the development process. The goal of V&V is to check if developed software meets the client's need & specification. V&V is a collection of testing and analysis activities across full life cycle. It determines that it performs its intended functions correctly. V&V uses structured approach to analyzes and test the hardware and software against all system functions. There are two branches of software testing. These are complementary to 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 validation without verification and vice versa cannot find a good software product. So we need combination of both.

Introduction V&V

Verification – Building the right software? It is the process for determining whether or not the product of a given phase of development fulfills the requirement established in the previous phases. For example suppose we have the specifications related to the project than by checking that specifications without executing to see whether the specifications are up to the mark or not is what we have done in verification

Validation – Are we building the right software? Validation is concerned with checking that the system will meet the customer’s actual needs, while verification is concerned with whether the system is well-engineered, error-free, and so on. Verification will help to determine whether the software is of high quality, but it will not ensure that the system is useful. Similarly Validation of the software is done to make sure that the software always meets the requirements of the customer by executing the specifications of the project and product.

Verification and Validation in Life Cycle
In this section V&V in SDLC mean what is checked in each phases and who check it. And we also that what techniques are used to perform the checking. V&V review it, analysis it and testing whether it meets the software requirements. Requirements include both quality attributes and functional capabilities. Its primary task is to identifying and monitoring errors in the project throughout the development and maintenance process. In the early stages it is impossible to identify all errors early in a project; it is the task of V&V to identify the error as soon as possible.

<table>
<thead>
<tr>
<th>Life Cycle Stages</th>
<th>Verification Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>1. Verification Approach determined</td>
</tr>
<tr>
<td></td>
<td>2. Requirement Adequacy Determined</td>
</tr>
<tr>
<td></td>
<td>3. Function Test Data Generated</td>
</tr>
<tr>
<td></td>
<td>4. Determine consistency of design with requirement</td>
</tr>
<tr>
<td>Design</td>
<td>1. Determine Adequacy of design</td>
</tr>
<tr>
<td></td>
<td>2. Create Functional test and Structure data</td>
</tr>
<tr>
<td></td>
<td>3. Determine consistency with design</td>
</tr>
<tr>
<td>Construction</td>
<td>1. Find Adequacy of Implementation</td>
</tr>
<tr>
<td></td>
<td>2. Create Functional test and Structure data for program</td>
</tr>
<tr>
<td></td>
<td>3. Apply Test Data</td>
</tr>
<tr>
<td>Operation Maintenance</td>
<td>1. Re- Verify, Commensurate with the level of redevelopment</td>
</tr>
</tbody>
</table>

Several types of Verification reviews are given below

1. Requirements review
2. Design review
3. Code walk through review
4. Code inspection
5. Test review
6. Verification

Verification

Is the product is designed according to needs of the customer. Verification is done at the starting of the development process. It includes reviews and meetings, walkthroughs, inspection, etc. to evaluate documents, plans, code, requirements and specifications. Suppose you are building a table. Here the verification is about checking all the parts of the table, whether all the four legs are of correct size or not. If one leg of table is not of the right size it wills imbalance the end product. Similar behavior is also noticed in case of the software product or application. If any feature of software product or application is not up to the mark or if any defect is found then it will result into the failure of the end product. Hence, verification is very important. It takes place at the starting of the development process.
Software verification and validation. It answers the questions like: Am I building the product right? Am I accessing the data right (in the right place; in the right way). It is a Low level activity Performed during development on key artifacts, like walkthroughs, reviews and inspections, mentor feedback, training, checklists and standards. Demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle. According to the Capability Maturity Model we can also define verification as the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

**Advantages of Software Verification:**

1. It helps in decreasing the number of the defect in the later stages of development.
2. When the product is verified at initial phases it helps to understand the product in a better way.
3. Chances of failures are reduced.
4. It also helps to build the product according to needs of customer.

**Methods of Verification**

**Self Review**

To improve the process it is necessary to keep the record and find the defects in self review. It is self review process. It is highly flexible in finding the defects, as one need not take an appointment for doing it.

**Peer Review**

A peer review is a developer who is supervisor with better knowledge and experience. It is conducted in various phases of SDLC. These are of two types.

1. **Online Review**
   
   In this method developer and reviewer meet together to review the work together

2. **Offline Review**
   
   In this method developer firstly informs to the user that your product is ready and then user gives the feedback to the developer. After getting feedback the developer proceed in the next stage.

**Walkthrough**

Walkthrough is a form of peer review in which developer leads the review process and other team member ask questions and spot possible errors against development standards. The meeting usually led by the author of the document under review and attended by other team member. The main purpose of walkthrough is to enable learning about the document under review to help team in understanding the contents of document.

**Inspection**

It is a formal review where external people involved as „inspector“. They are subject 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.

**Audits**

It is based on samples. These samples are conducted by auditors who may or may not experts. 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.

**Validation**

Determining whether it satisfies specified business requirements. After verification is completed Validation is done at the end of the development process a Am I building the right product? Am I accessing the right data (in terms of the data required to satisfy the requirement). It is a High level activity. Performed after a work
product is produced against established criteria ensuring that the product integrates correctly into the environment. Determination of correctness of the final software product by a development project with respect to the user needs and requirements.

Software verification and validation
According to the Capability Maturity Model we can also define validation as The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. A product can pass while verification, as it is done on the paper and no running or functional application is required. But, when same points which were verified on the paper is actually developed then the running application or product can fail while validation. This may happen because when a product or application is built as per the specification but these specifications is not up to the mark hence they fail to address the user requirements.

Advantages of Validation:
1. During verification if some defects are missed then during validation process it can be caught as failures.
2. If during verification some specification is misunderstood and development had happened then during validation process while executing that functionality the difference between the actual result and expected result can be understood.
3. Validation is done during testing like feature testing, integration testing, system testing, load testing, compatibility testing, stress testing, etc.
4. Validation helps in building the right product as per the customer’s requirement and helps in satisfying their needs.

Validation is basically done by the testers during the testing. While validating the product if some deviation is found in the actual result from the expected result then a bug is reported or an incident is raised. Not all incidents are bugs. But all bugs are incidents. Incidents can also be of type ‘Question’ where the functionality is not clear to the tester. Hence, validation helps in unfolding the exact functionality of the features and helps the testers to understand the product in much better way. It helps in making the product more users friendly.

Methods of Validation
Methods of validations are given below:

Unit Testing
It is a level of software testing where individual units/ components of software are tested. It is performed by developers. It is performed by white boxing method. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output.
Integration testing
In this method individual unit are combined and tested as a group. In this method developers support from independent test team. The purpose of this level is to expose faults in interaction between integrated units.

System testing
It is level of the software where complete and integrated software is tested. It is performed by independent test team. In this method testing of entire computer system is done. In this method usually black box testing is used.

Acceptance testing
In this method software is testing for acceptance of the software. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery. It is performed after system testing and before making the system available to the user.

Role of verification and Validation
Role of V&V is that it satisfies the need of the user so everything in the product requirement must be target of some V&V activity. According to the need of the user system must be developed otherwise it may lead failure at the end. V&V approaches described the functional and performance requirement and specification with respect to portability, safety, usability, security etc.

Improvement in Software Quality
Software quality interacts with each phase of 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. 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.

- 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.

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.

**VERIFICATION vs. VALIDATION**

The terms ‘Verification’ and ‘Validation’ are frequently used in the software testing world but the meaning of these terms are mostly vague and debatable. You will encounter (or have encountered) all kinds of usage and interpretations of those terms, and it is our humble attempt here to distinguish between them as clearly as possible.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Verification</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition</strong></td>
<td>Verification is the process of evaluating work-</td>
<td>Validation Testing ensures that the product</td>
</tr>
<tr>
<td></td>
<td>products of a development phase to determine</td>
<td>actually meets the client's needs. It can also</td>
</tr>
<tr>
<td></td>
<td>whether they meet the specified requirements.</td>
<td>be defined as to demonstrate that the product</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fulfills its intended use when deployed on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>appropriate environment.</td>
</tr>
<tr>
<td><strong>Objective</strong></td>
<td>To built the software according to the</td>
<td>To find that it meets the user need and that</td>
</tr>
<tr>
<td></td>
<td>requirements and design needs of user. In other</td>
<td>the specifications were correct in the first</td>
</tr>
<tr>
<td></td>
<td>words, to ensure that work products meet their</td>
<td>phase.</td>
</tr>
<tr>
<td></td>
<td>specified requirements.</td>
<td></td>
</tr>
<tr>
<td><strong>Question</strong></td>
<td>Are we building the product right?</td>
<td>Are we building the right product?</td>
</tr>
<tr>
<td><strong>Items</strong></td>
<td>Test Cases</td>
<td></td>
</tr>
<tr>
<td><strong>Activities</strong></td>
<td>1. Reviews</td>
<td>1. Testing</td>
</tr>
<tr>
<td></td>
<td>2. walkthroughs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Inspections</td>
<td></td>
</tr>
</tbody>
</table>

It is entirely possible that a product passes when verified but fails when validated. This can happen when, say, a product is built as per the specifications but the specifications themselves fail to address the user’s needs.

- Trust but Verify.
- Verify but also validate.

**Difference between Verification and Validation**

Verification and Validation example is also given just below to this table.
### Verification vs. Validation

<table>
<thead>
<tr>
<th>Verification</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. It is a static process of verifying documents, design, code and program.</td>
<td>1. It is a dynamic mechanism of validating and testing the actual product.</td>
</tr>
<tr>
<td>2. Code is not executed.</td>
<td>2. Code is executed.</td>
</tr>
<tr>
<td>4. It uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.</td>
<td>4. It uses methods like black box (functional) testing, gray box testing, and white box (structural) testing etc.</td>
</tr>
<tr>
<td>5. Whether the software conforms to specifications.</td>
<td>5. Whether software meets the customer expectations and requirements.</td>
</tr>
<tr>
<td>6. It is low level exercise So It can catch errors that validation cannot catch.</td>
<td>6. It is High Level Exercise It can catch errors that verification cannot catch.</td>
</tr>
<tr>
<td>7. Verification is done by Quality assurance team to ensure that the software is as per the specifications in the SRS document.</td>
<td>8. Validation is carried out with the involvement of testing team.</td>
</tr>
<tr>
<td>9. It generally comes first-done before validation.</td>
<td>9. It generally follows after verification.</td>
</tr>
</tbody>
</table>

### Conclusion and Perspective

So from above we say that V&V are essential part of SDLC, because they offer the only way to judge the quality and success of software. In V&V we must make 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
- www.softwaretestinghelp.com/
- softwaretestingfundamentals.com
- www.focus-itoutsourcing.com
- www.softwaretestingclass.com
- Software Verification and Validation: An Engineering and Scientific Approach, By Marcus S. Fisher
- Reference Information for the Software Verification and Validation Process, By Dolores R. Wallace, Laura M. Ippolito, Barbara B. Cuthill
- Software Verification and Validation for Practitioners and Managers, Steven R. Rakitin
- Verification, Validation and Testing in Software Engineering, edited by Aristides Dasso, Ana Funes
- Software Verification and Validation: A Practitioner's Guide, Steven R. Rakitin