

CHAPTER 2

VERIFICATION, VALIDATION, and V-Model

<u>VERIFICATION</u>	<u>VALIDATION</u>
1) It is a QA Activity	1) It is a QC Activity
2) Are we Building the Product right?	2) Are we Building the right Product?
3) Verification is the process of evaluating the work products of a software development lifecycle to check if we are in the right track of creating the final product.	3) Validation is the process of evaluating the final product to check whether the software meets the business needs or fulfils customers' needs.
4)) It is done by reading the documents manually	4) It is done by executing the software
5) It is also called as Static Testing	5) It is also called as Dynamic testing
6) It does not involve executing the code	6) It always involves executing the code
7) It is defect Preventive Approach	7) It is a defect detective and corrective approach
8) Here, the root cause(location) of the defect is found	8) Here, the root cause(location) of the defect is not found
9) The verifying process includes checking documents, design, code, and program	9) It is a dynamic mechanism of testing and validating the actual product
10) The Cost of Fixing the defect during verification is cheaper as compared to the cost of defect fixing during validation	10) The Cost of Fixing the defect during validation is costly as compared to the cost of defect fixing during verification
11) Verification happens at all phases of SDLC	11) Validation happens only at the last 3 phases of SDLC

12) Verification Techniques are:- 1) Walkthrough 2) Inspection 3) Review	12) Validation Techniques are:- (Levels of Testing) 1) Unit Testing 2) Integration Testing 3) System Testing 4) Acceptance Testing
13) It is done by the seniors of the respective phases	13) It is mostly done by the Testers

Verification Techniques:-

1) **Walkthrough:**

- It is an informal Process because here, Rules and Regulations is not followed, No seniors are involved , No proper Documentations that's why it is called as Informal Process
- It happens between the friends or colleagues.
- Walkthrough is just a casual talk between two or more colleagues were the colleague give their feedback or share the view-point regarding the work product to the author.
- Author leads the Walkthrough.

2) **Inspection :-**

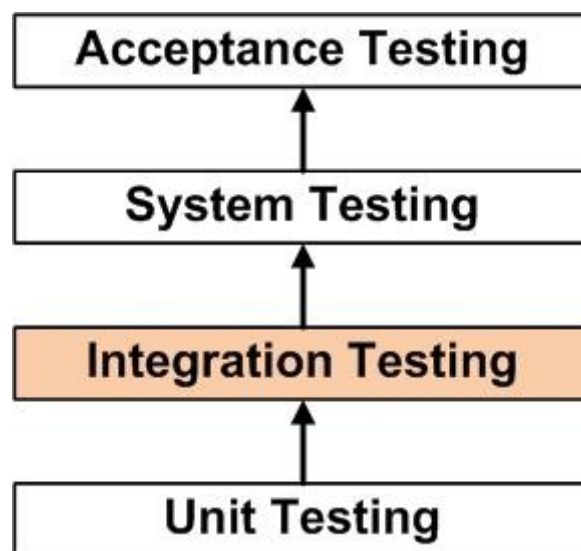
- It is Semi-formal process because here, Proper Rules and Regulation are followed, with Proper documentation, Seniors are involved but here the final Decision is not made that's why it is called Semi-formal process.
- There are 5 people who are involved in Inspection:-
 - Author:- the person who created the documents and whose documents is under inspection
 - Moderator:- the person who decides and inform all the people about the date, time and venue for the inspection.
 - Inspector/Reviewer:- the person who does the inspection of the product. They are the domain experts.
 - Reader:- the person who reads the documents for everyone in the inspection meeting
 - Recorder:- the person who note down all the events which occurred during the meeting, along with the defects details , corrections, and

follow-up details. In short. he/she makes a note of Minutes of meeting(MOM)

3) **Review:-**

- It is a formal Process because here the final decision is been taken that “whether the phase is complete and can we move to the next phase” and this decision is taken based on the correctness and completeness of the document.
- It is done by the senior most person (managers) of the project team.

Validation Techniques:-



1) **Unit Testing:-**

- Testing a small piece of executable code which gives the desirable output is called as Unit testing.
- It is done by the developers (White Box method is used for doing the unit testing). Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying every unit of the code used to implement functional requirements.
- “Unit testing is the method of verifying the smallest piece of testable code against its purpose.” If the purpose or requirement failed then the unit test has failed.
- In simple words, it means – writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.
- **UNIT TESTING** is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of

any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class.

2) **Integration Testing:-**

- It is done to test the communication link / Interfaces between two or more modules.
- When two or more modules are ready then integration testing is done on it.
- (What is modules:- collections of several units is called as one modules).
- **INTEGRATION TESTING** is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. For eg:- During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the ink cartridge and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed. For example, whether the cap fits into the body or not.
- **When is Integration Testing performed?:-** Integration Testing is the second level of testing performed after Unit Testing and before System Testing.
- **Who performs Integration Testing?:-** Developers themselves or independent testers perform Integration Testing.
- Point to remember before doing integration testing:-Ensure that you have a proper Detail Design document where interactions between each unit are clearly defined. In fact, you will not be able to perform Integration Testing without this information. Make sure that each unit is unit tested before you start Integration Testing.
- **There 3 approaches of doing Integration Testing:-**
 - i. **Top-Down Approach:-**
 1. When the parent module is ready and child module is not ready then Developers create a dummy code for the child module called as “**Test Stub**” for doing the integration testing.
 2. *Top Down* is an approach to Integration Testing where top-level units are tested first and lower level units are tested step by step after that.

This approach is taken when top-down development approach is followed. Test Stubs are needed to simulate lower level units which may not be available during the initial phases.

ii. **Bottom-Up Approach:-**

1. When the parent module is not ready and child module is ready then Developers create a dummy code for the Parent module called as “**Test Driver**” for doing the integration testing.
2. *Bottom Up* is an approach to Integration Testing where bottom level units are tested first and upper-level units step by step after that. This approach is taken when bottom-up development approach is followed. Test Drivers are needed to simulate higher level units which may not be available during the initial phases.

iii. **Big Bang Approach:-**

1. *Big Bang* is an approach to Integration Testing where all or most of the units are combined together and tested at one go. This approach is taken when the testing team receives the entire software in a bundle. So what is the difference between Big Bang Integration Testing and System Testing? Well, the former tests only the interactions between the units while the latter tests the entire system.
2. This Approach always leaves a doubt and consumes lots of time. That's why this approach is not used mostly for integration testing.

3) **System Testing:-**

- SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.
- The process of testing an integrated system to verify that it meets specified requirements.
- Here the software is tested with the scope of release whether the software is stable and completely eligible for releasing it to the client or not.
- **When is it performed?:-** System Testing is the third level of software testing performed after Integration Testing and before Acceptance Testing.
- **Who performs it?:-** Normally, independent Testers perform System Testing.
- In here both types of testing is carry forward – Functional Testing as well as Non-Functional Testing
- Testers use Black-Box Method for doing System testing.

- **System Testing** is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behaviour of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).

4) **Acceptance Testing:-**

- ACCEPTANCE TESTING is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.
- Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system. For eg:- During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the ink cartridge and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed. When the complete pen is integrated, System Testing is performed. Once System Testing is complete, Acceptance Testing is performed so as to confirm that the ballpoint pen is ready to be made available to the end-users.
- Acceptance Testing is done by the Customer or client or client representative.
- Here the customer takes the decision whether to accept the software or not.
- There are two types Acceptance Testing:-
 - i. **Alpha Testing:-**
 1. It is also called as Internal Testing / Factory testing.
 2. Here the users or customers come at the development environment(developer's site) and does the testing in presence of the development teams(programmer and testers).
 3. The identified bugs are resolved quickly by the development team there itself.
 - ii. **Beta Testing:-**
 1. It is also called as External testing / Field testing.
 2. Beta testing also known as user testing takes place at the Client environment(end users site) by the end users to validate the usability, functionality, compatibility, and reliability testing.

3. Beta testing adds value to the software development life cycle as it allows the "real" customer an opportunity to provide inputs into the design, functionality, and usability of a product.
4. In Case , if any defect(failure) found during the beta testing. The identified defect(failure) are reported to the development team for defect fixing.

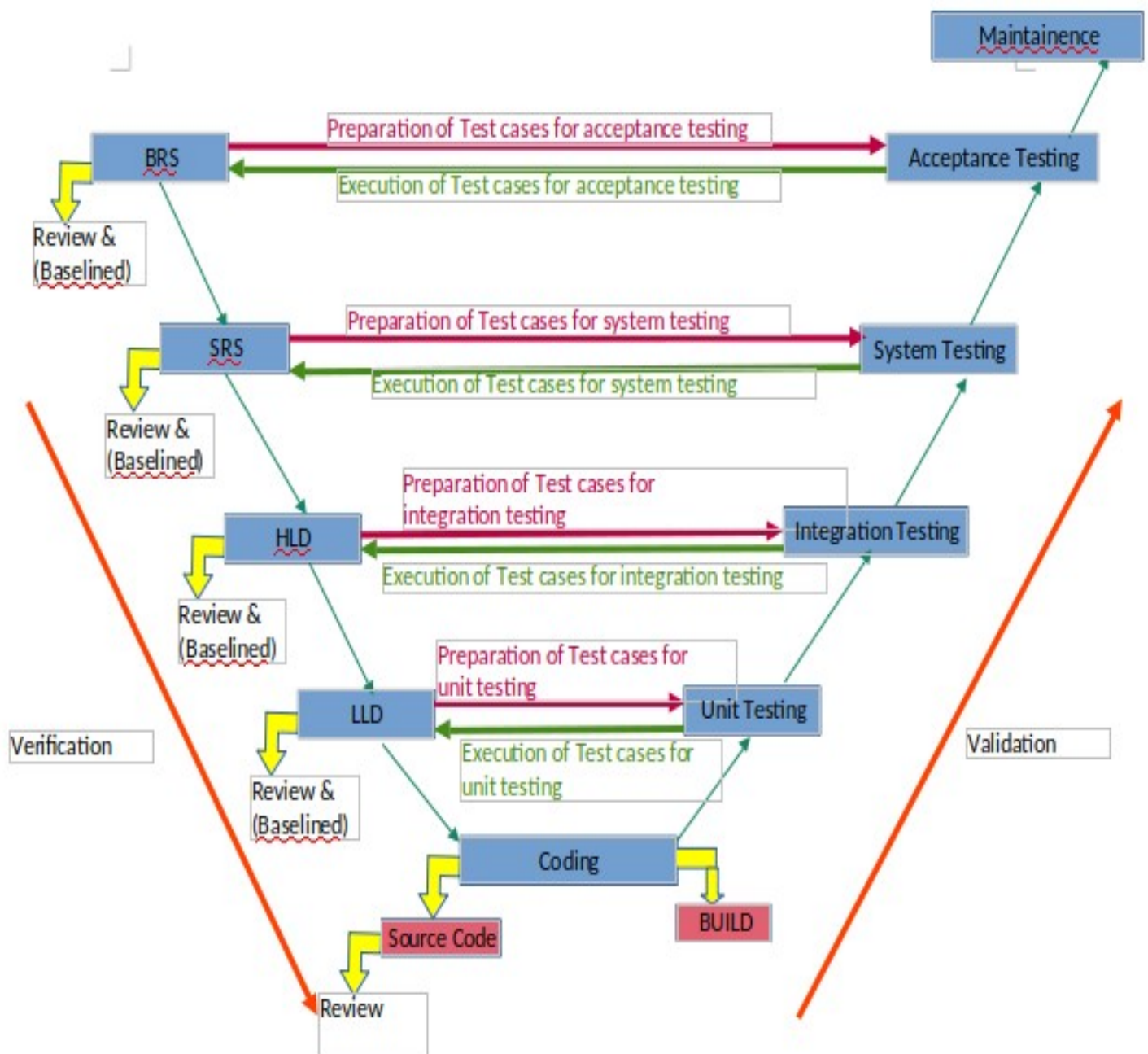
V-Model :-

When to use V Model?

V model is applicable when:

- Requirement is well defined and not ambiguous.
- Acceptance criteria are well defined.
- Project is short to medium in size.
- Technology and tools used are not dynamic.

<u>PROS(Advantages)</u>	<u>CONS(Disadvantages)</u>
<ul style="list-style-type: none">● Development and progress is very organized and systematic● Works well for smaller to medium sized projects.● Testing starts from beginning so ambiguities are identified from the beginning.● Easy to manage as each phase has well defined objectives and goals.● Simple and easy to use.● Avoids the downward flow of the defects.● Proactive defect tracking – that is defects are found at early stage.	<ul style="list-style-type: none">● Not suitable for bigger and complex projects● Not suitable if the requirements are not consistent.● No working software is produced in the intermediate stage.● No provision for doing risk analysis so uncertainty and risks are there.● Very rigid and least flexible.● Software is developed during the implementation phase, so no early prototypes of the software are produced.



V-Model