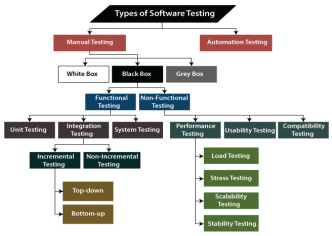
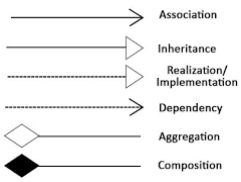
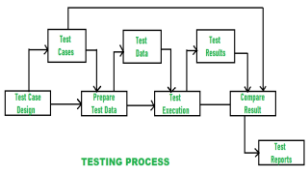
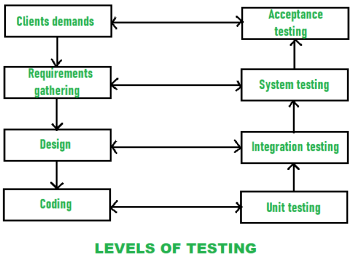






Software Design and Testing

Verification & Validation



Outline

- Verification
- Verification of Requirements
- High-level and Low-Level Design
- How to Verify Code?
- Validation
- Validation Activities
- Unit Testing
- Integration Testing
- Function Testing
- System Testing
- Acceptance Testing
- Overview of Regression Testing

Introduction

- Every validation testing focuses on a particular stage of SDLC phase and on a particular class of errors
- So there is a one-to-one correspondence between development and testing processes
- E.g. the purpose of system validation testing is to explore whether the product is consistent with the original objectives
- Advantage:
  - This structure of validation testing avoids redundant testing and prevents one from overlooking large classes of errors.
  - s/w validation is achieved through a series of black-box tests that demonstrate conformity with requirements.
- A test plan outlines the classes of tests to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with requirements.

3

Introduction

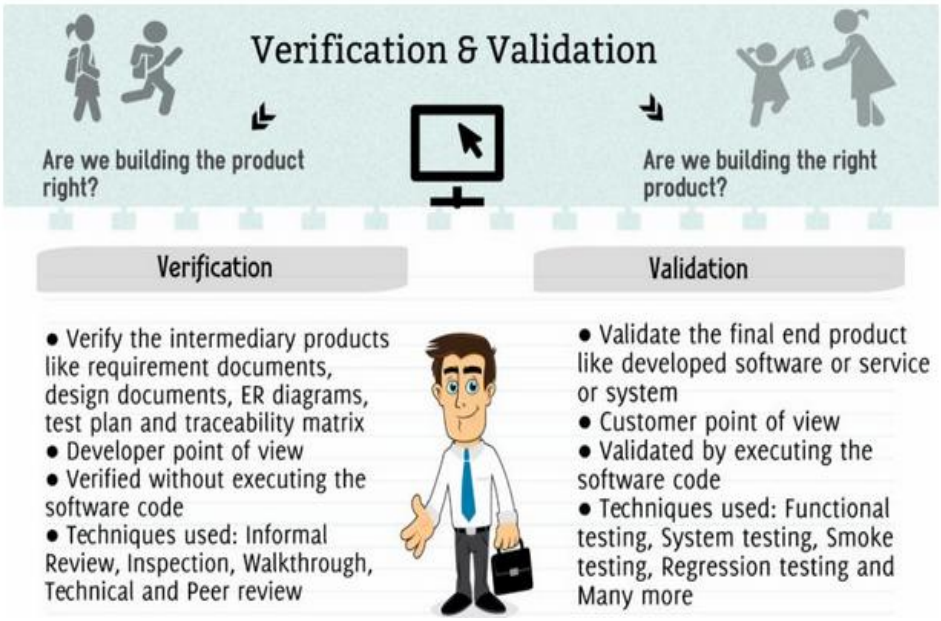
- The goal of plan and procedure:
  - All functional requirements are satisfied
  - All behavioral characteristics are achieved
  - All performance requirements are attained
  - Documentation is correct
  - Human-engineered and other requirements are met

4

Development Of Test Strategy

- You can divide a complex task into many sub-tasks. Every sub-task is developed and accomplished towards achieving the complex task.
- Checking every sub-task to ensure that you are working in a right direction. This is **verification**.
- After sub-tasks have been completed and merged, the entire task is checked to ensure the required task goals have been achieved. This is **validation**.
- **Verification Ensures that**: Every Step in the Process of Building the Software Delivers the Correct Product.
- **Validation Ensures that**: Software Being Developed or Changed Satisfies Functional and All Other Requirements.

5

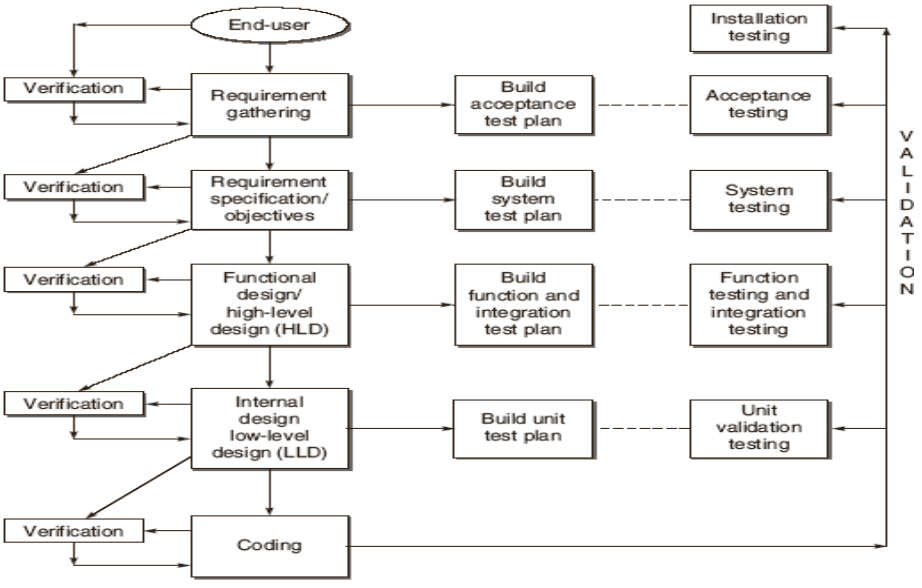


Development Of Test Strategy

- Verification is checking the work at intermediate level to confirm that the project is moving in the right direction, towards the set goal.
- When a module is prepared with various stages of SDLC like plan, design and code, it is verified at every stage.
- But there may be more than one modules in the system which need to be integrated.
- Therefore after building individual modules following stages need to be tested: the module as a whole, integration of modules, and the system built after integration – this is called validation testing.

7

V & V Activities

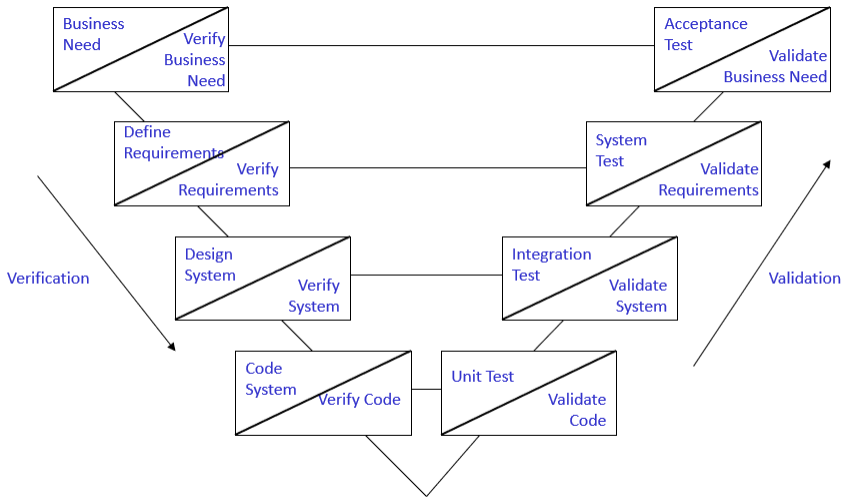


Testing Life Cycle Model

- V & V are the building blocks of a testing process on which the testing strategy is based.
- This model is known as the testing life cycle model. Life cycle involves continuous testing of the system during the development process
- Life cycle testing is dependent on the completion of predetermined deliverables at a specified point in the development life cycle
- In V-testing when the project starts, both the system development and the testing begin
- As soon as the development starts, the testing team begins planning the system test process as shown in the following figure.

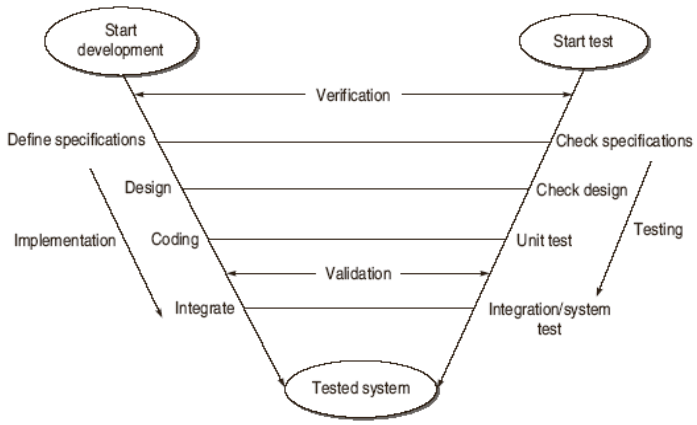
9

V'Life Cycle Model



10

V'Life Cycle Model



- The V & V process in nutshell, involves (i) verification of every step of SDLC and (ii) validation of the verified at the end.

11

V-Testing

- A V-diagram provides the following insights about s/w testing:
  - Testing can be implemented in the same flow as for SDLC
  - Testing can be broadly planned in two activities, namely verification and validation
  - Testing must be performed at every step of SDLC
  - V-diagram supports the concept of early testing
  - V-diagram supports parallelism in the activities of the developers and testers
  - The more you concentrate in the V & V process, more will be the cost-effectiveness of the s/w
  - Testers should be involved in the development process

12

V-Testing

- A V-diagram provides the following insights about s/w testing:
  - Testing can be implemented in the same flow as for SDLC
  - Testing can be broadly planned in two activities, namely verification and validation
  - Testing must be performed at every step of SDLC
  - V-diagram supports the concept of early testing
  - V-diagram supports parallelism in the activities of the developers and testers
  - The more you concentrate in the V & V process, more will be the cost-effectiveness of the s/w
  - Testers should be involved in the development process

13

Differences between verification and validation

	Verification	Validation
Definition	It is a process of checking if a product is developed as per the specifications.	It is a process of ensuring that the product meets the needs and expectations of stakeholders.
What it tests or checks for	It tests the requirements, architecture, design, and code of the software product.	It tests the usability, functionalities, and reliability of the end product.

14

Differences between verification and validation

	Verification	Validation
Coding requirement	It does not require executing the code.	It emphasizes executing the code to test the usability and functionality of the end product.
Activities include	A few activities involved in verification testing are requirements verification, design verification, and code verification.	The commonly-used validation activities in software testing are usability testing, performance testing, system testing, security testing, and functionality testing.

15

Differences between verification and validation

	Verification	Validation
Types of testing methods	A few verification methods are inspection, code review, desk-checking, and walkthroughs.	A few widely-used validation methods are black box testing, white box testing, integration testing, and acceptance testing.
Teams or persons involved	The quality assurance (QA) team would be engaged in the verification process.	The software testing team along with the QA team would be engaged in the validation process
Target of test	It targets internal aspects such as requirements, design, software architecture, database, and code.	It targets the end product that is ready to be deployed.

16

V&V Activities

- V & V activities can be understood using SDLC phases as follows:
  - Phases: End user -> Requirement gathering -> Requirement specifications/ objectives -> Functional design/high-level design(HLD) -> Internal Design/low-level design(LLD) -> Coding.
- **Requirement gathering**: the requirements gathered from the user's viewpoints only with no technical details are translated into a written set of requirements
- **Requirement specifications/ objectives**: the user's requirements are specified in developer's terminology and the specified objectives are created known as SRS
- **FD & HLD**: FD is the process of translating user requirements into a set of external interfaces. HLD is prepared with SRS and s/w analysts convert the requirements into a usable product

17

V&V Activities

- Thus, a HLD document will contain following items at macro level:
  - Overall architecture diagram along with technology details
  - Functionalities of the overall system with the set of external interfaces
  - List of modules
  - Brief functionalities of each module
  - Interface relationship among modules including dependencies between modules, database tables identified along with key elements

18

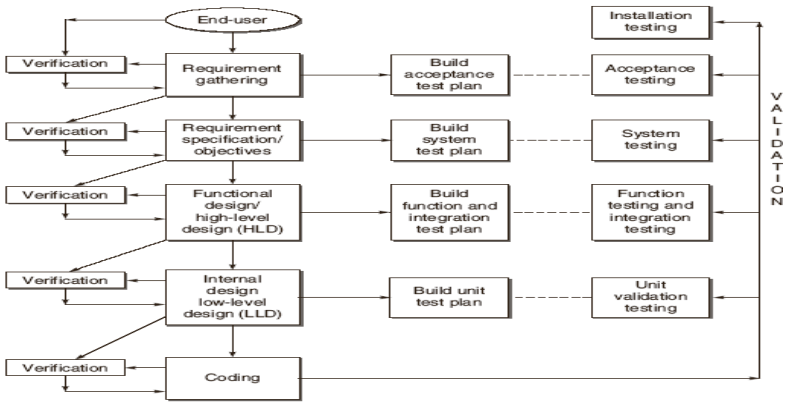
V&V Activities

- **Internal Design or LLD**: HLD cannot be given to the programmers for coding as it contains macro-level details only
  - So a micro-level design document called as internal design or low-level design is prepared with elaborate description of each module.
  - There should be at least one separate document for each module
- **Coding**: Here coding is done using the design document for a module.
  - After all the SDLC phases, we need to put together all the verification activities as it is performed at all phases
  - Along with the verification activities performed at every step, the tester needs to prepare some test plans which will be used in validation activities performed after coding the system

19

V&V Activities

- When the coding is over for a unit or a system, and parallel verification activities have been performed, then the system can be validated.
- These are executed with the help of test plans prepared by the testers at every phase of SDLC (below figure).



20

Verification

- Verification ensures correct implementation of specific functions in a s/w
- What is the need of verification? Cant we just test the s/w in the final phase of SDLC?
  - If verification is not performed at early stages, there are always chances of mismatch between the required product and the delivered product.
  - E.g. if requirements are not verified , it may lead to something where there are not clear with commitments
  - Verification exposes more errors
  - Early verification decreases the cost of fixing bugs
  - Early verification enhances the quality of the s/w

21

Examples of V & V

- A clickable button with name Submet.
- Verification would check the design doc and correcting the spelling mistake.
- Otherwise, the development team will create a button like

Example of Verification

22

Examples of V & V

- So new specification is
- A clickable button with name Submit
- Once the code is ready, Validation is done. A Validation test found –

Example of Validation

- Owing to validation testing, the development team will make the submit button clickable.

23

Goals of Verification

- After understanding the need of verification , the goals must be verified as follows:
- **Everything Must Be Verified**
  - In principle, all the SDLC phases and all the products of these processes must be verified
  - Results of Verification May Not Be Binary
  - Verification may not be just the acceptance or rejection of a product
  - Often, one has to accept approximations
  - E.g sometimes correctness of the requirements cannot be rejected or accepted outright but can be accepted with a degree of satisfaction or rejected with the degree of modification

24

Goals of Verification

- **Even Implicit Qualities Must Be Verified**
  - The qualities in the s/w are explicitly stated in the SRS
  - But those requirements which are implicit and not mentioned anywhere must also be verified

25

Verification Activities

- All verification activities are performed in connection with the different phases of SDLC as follows:
  - Verification of Requirements ad Objectives
  - Verification of High-Level Design
  - Verification of Low-Level Design
  - Verification of Coding(Unit Verification)

26