Software Engineering

Prepared by: Neha Tripathi

SDLC- V Model

- also called as *Verification and Validation Model*

V-Model

- The V-model is a type of SDLC model where process executes in a sequential manner in V-shape.

- It is also known as Verification and Validation model.

- It is based on the association of a testing phase for each corresponding development stage.

- Development of each step directly associated with the testing phase.

- The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.

V-model representation
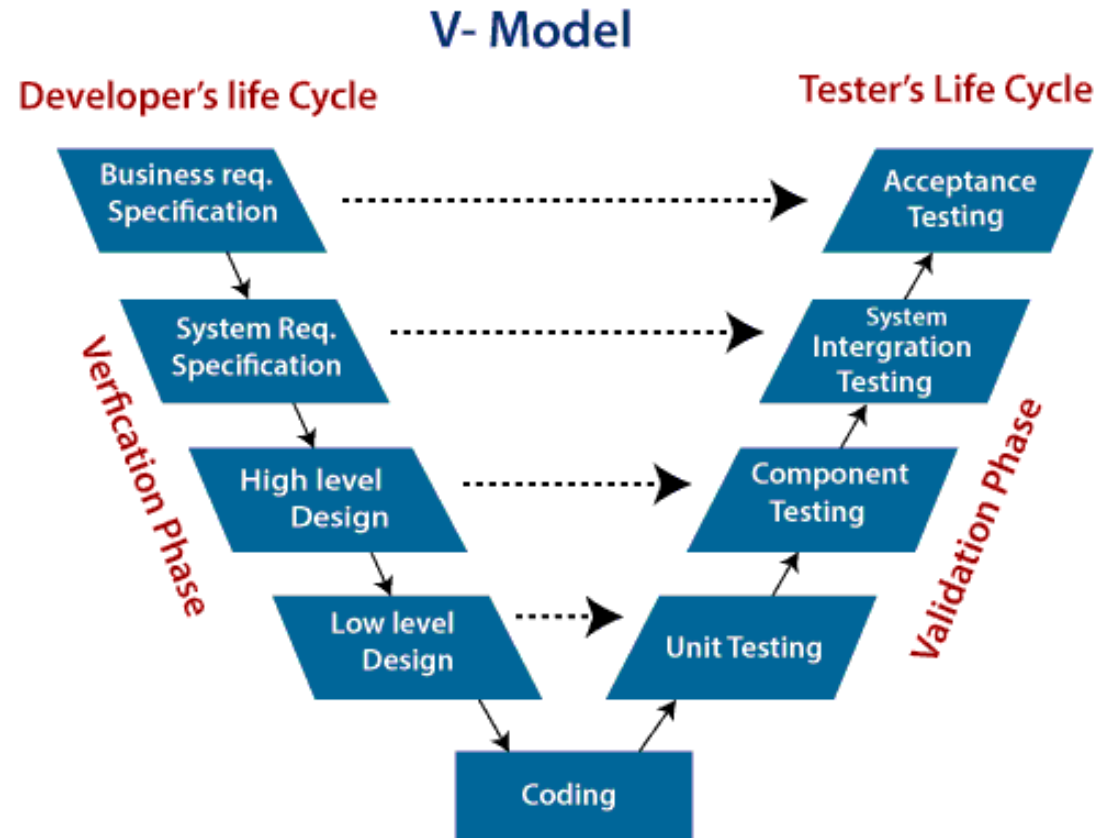
V-Model representing the document released after each phase that assist in corresponding testing
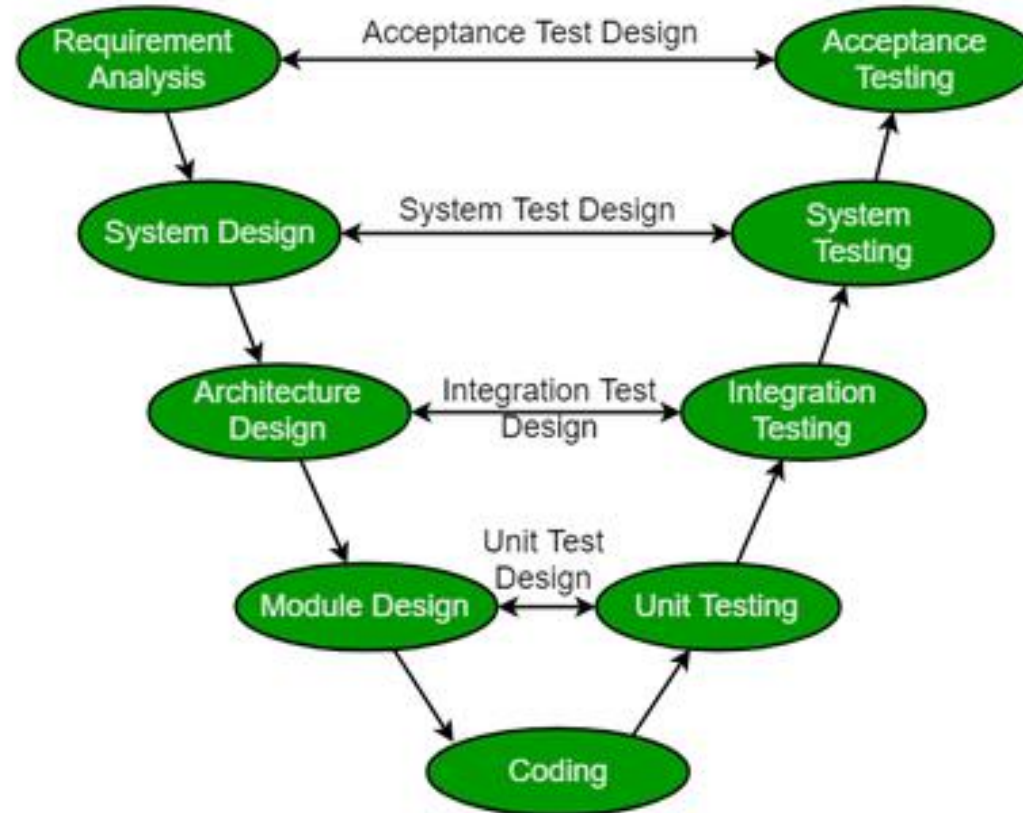
Continue…

- **Verification:** It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet.

- **Validation:** It involves dynamic analysis technique (functional, non-functional), testing done by executing code. Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements.

- **So V-Model contains Verification phases on one side of the Validation phases on the other side. Verification and Validation phases are joined by coding phase in V-shape. Thus it is called V-Model.**

Verification Phase of V-model:

- **Requirement Analysis:** This phase contains detailed communication with the customer to understand their requirements and expectations. This stage is known as Requirement Gathering.

- **System Design:** This phase contains the system design and the complete hardware and communication setup for developing product.

- **Architectural Design:** System design is broken down further into modules taking up different functionalities. The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood.

- **Module Design:** In this phase the system breaks down into small modules. The detailed design of modules is specified, also known as Low-Level Design (LLD).

Validation Phase of V-model:

- **Unit Testing:** Unit Test Plans are developed during module design phase. These Unit Test Plans are executed to eliminate bugs at code or unit level.

- **Integration testing:** After completion of unit testing Integration testing is performed. In integration testing, the modules are integrated and the system is tested. Integration testing is performed on the Architecture design phase. This test verifies the communication of modules among themselves.

- **System Testing:** System testing test the complete application with its functionality, inter dependency, and communication . It tests the functional and non-functional requirements of the developed application.

- **User Acceptance Testing (UAT):** UAT is performed in a user environment that resembles the production environment. UAT verifies that the delivered system meets user's requirement and system is ready for use in real world.

Principles of V-Model:

- **Large to Small:** In V-Model, testing is done in a hierarchical perspective, For example, requirements identified by the project team, create High-Level Design, and Detailed Design phases of the project. As each of these phases is completed the requirements, they are defining become more and more refined and detailed.

- **Data/Process Integrity:** This principle states that the successful design of any project requires the incorporation and cohesion of both data and processes. Process elements must be identified at each and every requirements.

- **Scalability:** This principle states that the V-Model concept has the flexibility to accommodate any IT project irrespective of its size, complexity or duration.

- **Cross Referencing:** Direct correlation between requirements and corresponding testing activity is known as cross-referencing.

- **Tangible Documentation:** This principle states that every project needs to create a document. This documentation is required and applied by both the project development team and the support team. Documentation is used to maintaining the application once it is available in a production environment.

Why preferred?

- It is easy to manage due to the rigidity of the model. Each phase of V-Model has specific deliverables and a review process.

- Proactive defect tracking – that is defects are found at early stage.

When to use V-Model?

- When the requirement is well defined and not ambiguous.

- The V-shaped model should be used for small to medium-sized projects where requirements are clearly defined and fixed.

- The V-shaped model should be chosen when sample technical resources are available with essential technical expertise.

Advantages:

- Easy to Understand.

- Testing Methods like planning, test designing happens well before coding.

- This saves a lot of time. Hence a higher chance of success over the waterfall model.

- Avoids the downward flow of the defects.

- Works well for small plans where requirements are easily understood.

Disadvantages:

- Very rigid and least flexible.

- Not a good for a complex project.

- Software is developed during the implementation stage, so no early prototypes of the software are produced.

- If any changes happen in the midway, then the test documents along with the required documents, has to be updated.

Thank You!