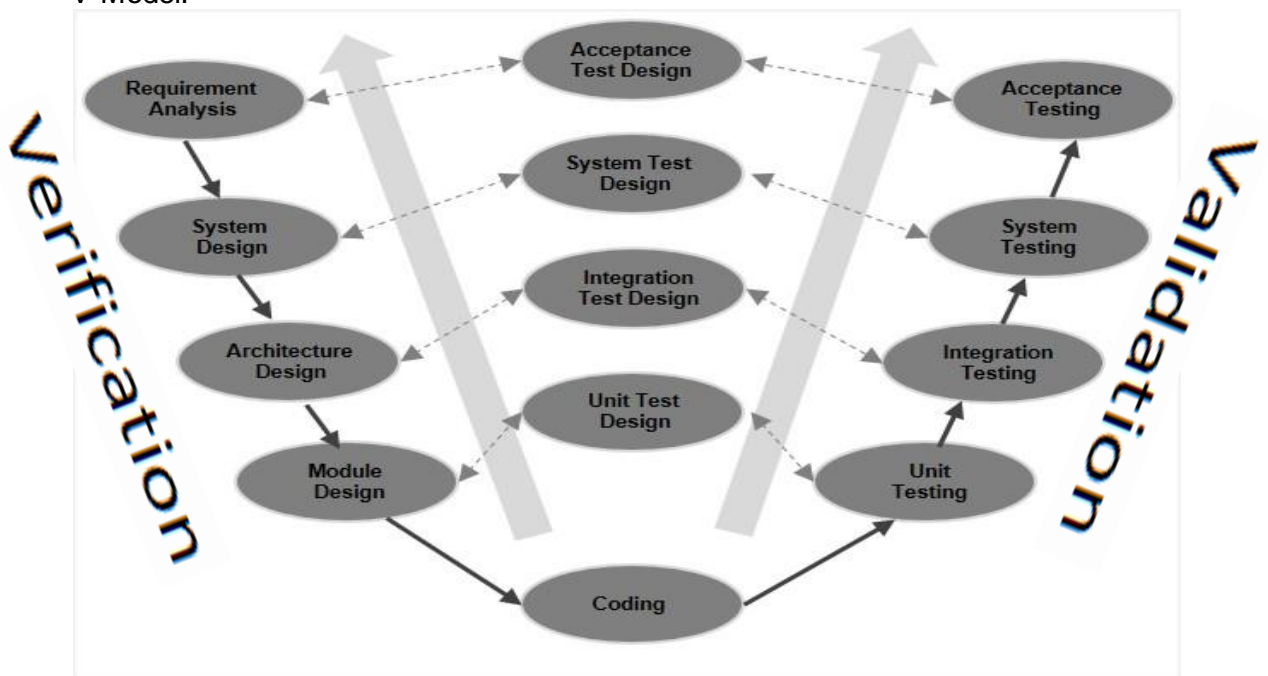


# V MODEL

- **Paul Rook in 1980** invented **V-Model**.
- The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as **Verification and Validation model**.
- The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly disciplined model and the next phase starts only after completion of the previous phase.
- Under the V-Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the 'V' and Validation phases on the other side. The Coding Phase joins the two sides of the V-Model.



## V-Model - Verification Phases

- **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.
- **Architecture Design:** Architectural specifications are understood and designed in this phase. The system design is broken down further into modules taking up different functionality. This is also referred to as **High Level Design (HLD)**. The **high-level design (HLD)** phase focuses on system architecture and design. It provides overview of solution, platform, system, product and service/process.

An **integration test** plan is created in this phase as well in order to test the pieces of the software systems ability to work together.

- **Module Design:** In this phase, the detailed internal design for all the system modules is specified, referred to as **Low Level Design (LLD)**. The low-level design (LLD) phase is where the actual software components are designed. It defines the actual logic for each and every component of the system. Class diagram with all the methods and relation between classes comes under LLD. Component tests are created in this phase as well.
- **Coding:** The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements. The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

## V-Model - Validation Phases

- **Unit Testing:** Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing. Unit testing is applied on the basis of unit test design.
- **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. Integration testing is applied on the basis of integration test design.
- **System Testing:** System testing is directly associated with the system design phase. System tests check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware compatibility issues can be uncovered during this system test execution. System testing is applied on the basis of system test design.
- **Acceptance Testing:** Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment. Acceptance tests uncover the compatibility issues with the other systems available in the user environment. It also discovers the non-functional issues such as load and performance defects in the actual user environment. Acceptance testing is applied on the basis of acceptance test design.

## V- Model — Application

V- Model application is almost the same as the waterfall model, as both the models are of sequential type. Requirements have to be very clear before the project starts, because it is usually expensive to go back and make changes. This model is used in the medical development field, as it is strictly a disciplined domain.

The following pointers are some of the most suitable scenarios to use the V-Model application.

- Requirements are well defined, clearly documented, and fixed.
- Product definition is stable.
- Technology is not dynamic and is well understood by the project team.
- There are no ambiguous or undefined requirements.
- The project is short

## V - Model - Advantages

The advantage of the V-Model method is that it is very easy to understand and apply. The simplicity of this model also makes it easier to manage.

The advantages of the V-Model method are as follows –

- This is a highly disciplined model and Phases are completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.

## V - Model – Disadvantages

The disadvantage is that the model is not flexible to changes and just in case there is a requirement change, which is very common in today's dynamic world, it becomes very expensive to make the change

The disadvantages of the V-Model method are as follows –

- High risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

- Once an application is in the testing stage, it is difficult to go back and change a functionality.
- No working software is produced until late during the life cycle.