V MODEL

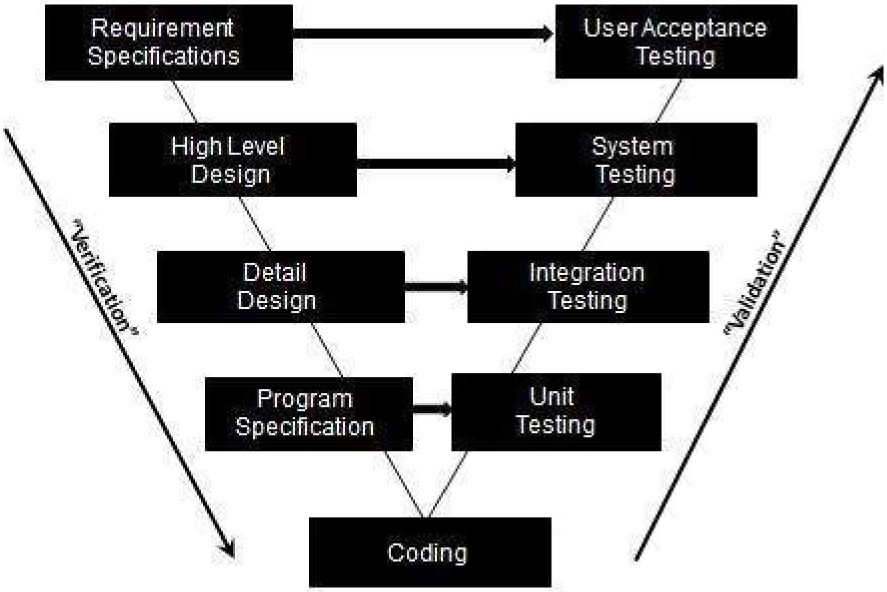
V Model - SDLC:

V model, a software development life cycle methodology, describes the activities to be performed and the results that have to be produced during the life cycle of the product. It is known as verification and validation model Validation answers the question – "Are we developing the product which attempts all that user needs from this software ?" and Verification answers the question– "Are we developing this product by firmly following all design specifications ?"

V-Model Objectives:

- Project Risks Minimization
- Guaranteed Quality
- Total Cost reduction of the Entire Project
- Improved Communication between all Parties Involved

V-Model Different Phases:



- **The Requirements phase**, a document describing what the software is required to do after the software is gathered and analyzed and the corresponding test activity is **user acceptance testing**.
- **The Architectural Design phase**, where a software architecture is designed and building the components within the software and the establishing the relationships between the components and the corresponding test activity is System Testing.
- **The High Level Design phase,** breaking the system into subsystems with identified interfaces; then gets translated to a more detailed design and the corresponding test activity is Integration testing.
- **The Detailed Design phase,** where the detailed implementation of each component is specified. The detailed design broken into Data structures, Algorithm used and the corresponding test activity is unit Testing.
- **Coding** in which each component of the software is coded and tested to verify if faithfully implements the detailed design.

Advantages and Limitations of V-Model:

Advantages:

- Emphasize for verification and validation of the product in early stages of product development.
- Each stage is testable
- Project management can track progress by milestones
- Easy to understand implement and use

Limitations:

- Does not easily handle events concurrently.
- Does not handle iterations or phases
- Does not easily handle dynamic changes in requirements
- Does not contain risk analysis or Mitigation activities