V-Model

The V-Model, as depicted in the diagram, represents a software development lifecycle model emphasizing a structured and sequential development and testing process. **It is called the "V-Model" because of its V-shaped structure**, where the left side represents the **verification** phase (developer's lifecycle), and the right side represents the **validation phase (tester's lifecycle). The center of the "V" is the coding phase**, which bridges the two halves.

Here's a detailed explanation of each stage in the diagram:

Verification Phase (Left Side of the V)

This phase focuses on planning, designing, and verifying that the product is being built correctly according to requirements.

1. **Business Requirement Specification (BRS)**
    - Description: Defines the **high-level business goals and user needs.**
    - Output: **A business requirement document (BRD).**
    - Corresponding Test Activity: **Acceptance Testing**.
2. **System Requirement Specification (SRS)**
    - Description: Elaborates on **detailed functional and non-functional requirements**.
    - Output: **A system requirement specification document**.
    - Corresponding Test Activity: System Integration Testing.
3. **High-Level Design (HLD)**
    - Description: Describes the **architecture and design of the system**, **breaking down modules and their interactions.**
    - Output: High-level design documents.
    - Corresponding Test Activity: Component Testing.
4. **Low-Level Design (LLD)**
    - Description: Focuses **on detailed designs for each module or component**.
    - Output: Low-level design documents.
    - Corresponding Test Activity: **Unit Testing**.
5. **Coding**
    - Description: The actual development of the software system, converting designs into executable code.

Validation Phase (Right Side of the V)

This phase involves testing to ensure the product meets the business and technical requirements.

1. **Unit Testing**
    - **Tests individual components or modules as defined in the LLD**.
    - Ensures each unit functions as intended.
2. **Component Testing**
    - Verifies the **integration and interaction between multiple components.**
    - Ensures data flow between modules works as expected.
3. **System Integration Testing**
    - Validates **the overall system's functionality** against the **system requirements**.
    - Ensures the system works in an integrated environment.
4. **Acceptance Testing**
    - Confirms that the **final product meets the business requirements (BRS).**
    - Often performed by the client or end-users.
