ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY
UNIVERSITY FOR INDUSTRY

Department of Software Engineering

College of Engineering

Course Name: Software Component Design

Section A

TITLE: - V-model software development

NAME	ID No
Aduna Kebede	ETS0088/13
Abel Bekele	ETS0022/13
Betselot Bezuayehu	ETS0252/13
Bethelhem Gebeyehu	ETS0243/13
Bethelhem Habtamu	ETS0244/13

Submission Date: Dec, 18, 2024

Introduction

The V-Model, also known as the Verification and Validation Model, is a structured software development life cycle (SDLC) model that provides a systematic and visual representation of the software development process. It is based on the idea of a "V" shape, with the two legs of the "V" representing the progression of the software development process from requirements gathering and analysis to design, implementation, testing, and maintenance. The V-Model is widely used in industries where safety and reliability are critical, such as aerospace, defense, and healthcare.

The following illustration depicts the different phases in a V-Model of the SDLC.

Verification Phases:

It involves a static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements are met.

There are several Verification phases in the V-Model:

Business Requirement Analysis:

This is the first step of the designation of the development cycle where product requirement needs to be cured from the customer's perspective. in these phases include proper communication with the customer to understand the requirements of the customers. these are the very important activities that need to be handled properly, as most of the time customers do not know exactly what they want, and they are not sure about it at that time then we use an **acceptance test design** planning which is done at the time of business requirement it will be used as an **input** for acceptance testing.

System Design:

Design of the system will start when the overall we are clear with the product requirements, and then need to design the system completely. This understanding will be at the beginning of complete under the product development process. these will be beneficial for the future execution of test cases.

Architectural Design:

In this stage, architectural specifications are comprehended and designed. Usually, several technical approaches are put out, and the ultimate choice is made after considering both the technical and financial viability. The system architecture is further divided into modules that each handle a distinct function. Another name for this is High-Level Design (HLD).

Module Design:

This phase, known as Low-Level Design (LLD), specifies the comprehensive internal design for every system module. Compatibility between the design and other external systems as well as other modules in the system architecture is crucial. Unit tests are a crucial component of any development process

since they assist in identifying and eradicating the majority of mistakes and flaws at an early stage. Based on the internal module designs, these unit tests may now be created.

Coding Phase:

The Coding step involves writing the code for the system modules that were created during the Design phase. The system and architectural requirements are used to determine which programming language is most appropriate.

The coding standards and principles are followed when performing the coding. Before the final build is checked into the repository, the code undergoes many code reviews and is optimized for optimal performance.

Validation Phases:

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

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

Unit Testing:

Unit Test Plans are developed during the module design phase. These Unit Test Plans are executed to eliminate bugs in 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 in the Architecture design phase. This test verifies the communication of modules among themselves.

System Testing:

System testing tests 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 the user's requirement and the system is ready for use in the real world.

Design Phase:

- **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 the 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).

Testing Phases:

- **Unit Testing:** Unit Test Plans are developed during the module design phase. These Unit Test Plans are executed to eliminate bugs at the 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 in the Architecture design phase. This test verifies the communication of modules among themselves.
- **System Testing:** System testing tests the complete application with its functionality, interdependency, and communication. It tests the functional and non-functional requirements of the developed application.

Importance of V-Model

1. Early Defect Identification

By incorporating verification and validation tasks into every stage of the development process, the V-Model encourages early testing. This lowers the cost and effort needed to remedy problems later in the development lifecycle by assisting in the early detection and resolution of faults.

2. determining the Phases of Development and Testing

The V-Model contains a testing phase that corresponds to each stage of the development process. By ensuring that testing and development processes are clearly mapped out, this clear mapping promotes a methodical and orderly approach to software engineering.

3. Prevents “Big Bang” Testing

Testing is frequently done at the very end of the development lifecycle in traditional development models, which results in a “Big Bang” approach where all testing operations are focused at once. By integrating testing activities into the development process and encouraging a more progressive and regulated testing approach, the V-Model prevents this.

4. Improves Cooperation

At every level, the V-Model promotes cooperation between the testing and development teams. Through this collaboration, project requirements, design choices, and testing methodologies are better understood, which improves the effectiveness and efficiency of the development process.

5. Improved Quality Assurance

Overall quality assurance is enhanced by the V-Model, which incorporates testing operations at every level. Before the program reaches the final deployment stage, it makes sure that it satisfies the requirements and goes through a strict validation and verification process.

When to Use of V-Model?

- **Traceability of Requirements:** The V-Model proves beneficial in situations when it's imperative to create precise traceability between the requirements and their related test cases.
- **Complex Projects:** The V-Model offers a methodical way to manage testing activities and reduce risks related to integration and interface problems for projects with a high level of complexity and interdependencies among system components.
- **Waterfall-Like Projects**: Since the V-Model offers an approachable structure for organizing, carrying out, and monitoring testing activities at every level of development, it is appropriate for projects that use a sequential approach to development, much like the waterfall model.

Advantages of V-Model

- This is a highly disciplined model and Phases are completed one at a time.
- V-Model is used for small projects where project requirements are clear.
- Simple and easy to understand and use.
- This model focuses on verification and validation activities early in the life cycle thereby enhancing the probability of building an error-free and good quality product.

Disadvantages of V-Model

- High risk and uncertainty.
- It is not good for complex and object-oriented projects.
- It is not suitable for projects where requirements are not clear and contain a high risk of changing.
- This model does not support iteration of phases.
- Time-Consuming: The V-Model can be time-consuming, as it requires a lot of documentation and testing.