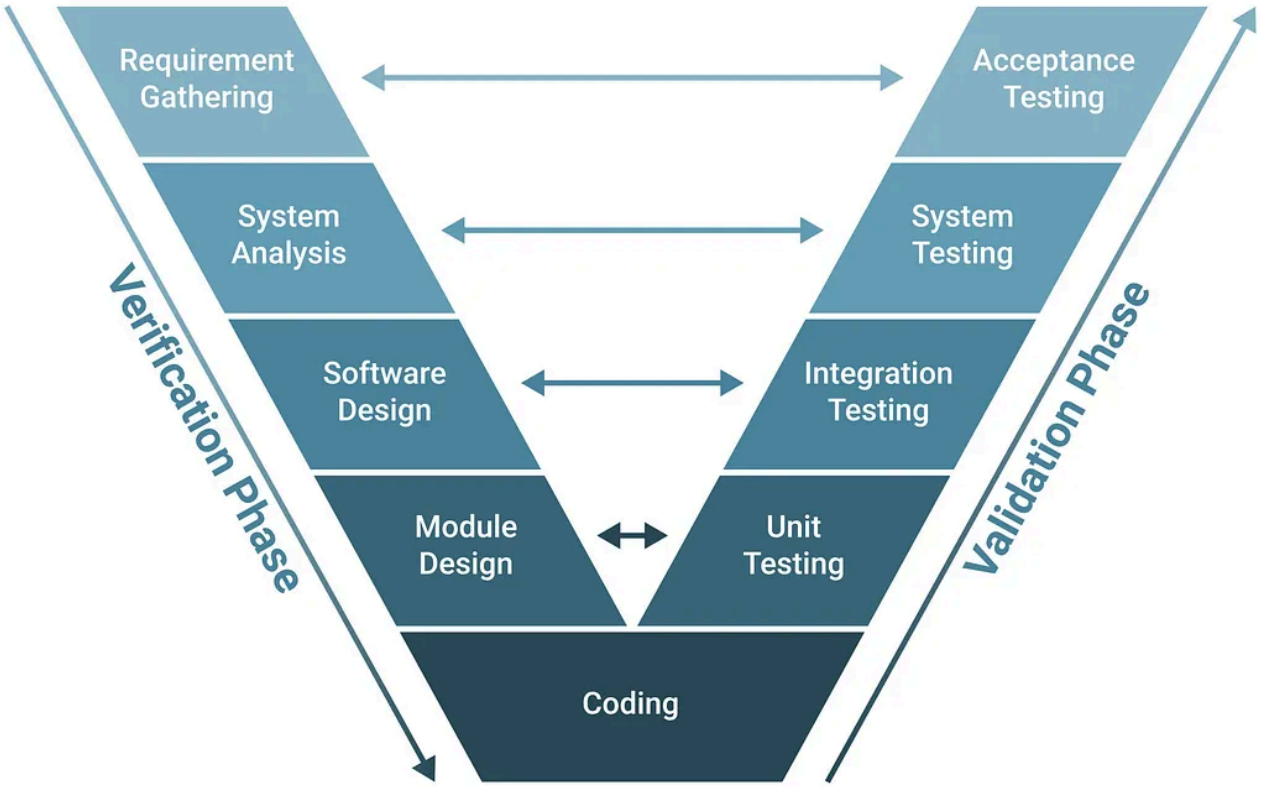
The V-Model in Software Testing: A Simple Guide to Structured Quality

Rashmi Ruwanthika Athukorala 3 min read · Just now

The **V-Model** is a method used in **software development** that helps make sure both development and testing happen together.
Unlike the **Waterfall model**, where testing happens only after the development is finished, the V-Model makes sure testing is planned from the very beginning.

This means fewer mistakes, better quality, and less money spent fixing bugs later.
It is also called the **Verification and Validation Model**, and it is named the "V-Model" because the process looks like the letter **V** when drawn as a diagram.



🏗️ Understanding the Structure: Verification and Validation

The V-Model has two main sides:

- **Left side:** Verification (Development work — building the product correctly)
- **Right side:** Validation (Testing work — checking if the right product was built)

Let's look at each side step by step 👇

🔷 The Left Side: Verification (Development Phases)

This side represents the steps where software is designed and developed. Each step has a matching testing stage on the right side of the “V”.

1. **Business Requirement Analysis**

- This is the first step where we understand what the user or client wants,
- Example: “The user should be able to log in and view their dashboard.”

2. System Design

- Here, we decide the main structure of the system — how it will work as a whole.
- Example: Choosing the database, deciding how modules connect, etc.

3. Architectural Design (High-Level Design)

- The system is divided into main parts or modules.
- Example: Login module, payment module, and report modul.

4. Module Design (Low-Level Design)

- Each module is planned in detail — what each function or button should do.

5. Coding

- Developers start writing the actual program based on all the designs.
- This is the bottom point of the “V,” where design turns into real code.

The Right Side: Validation (Testing Phases)

The right side represents the testing process — where we check that each part of the software works properly.
Each testing phase matches a development phase from the left side.

Development Phase	↔	Testing Phase
Business Requirement Analysis	↔	User Acceptance Testing (UAT)
System Design	↔	System Testing
Architectural Design	↔	Integration Testing
Module Design	↔	Unit Testing

The Four Testing Levels Explained

1. **Unit Testing**

- Done by developers.
- Tests one small part (a function or module) to make sure it works correctly.

2. Integration Testing

- Checks if different modules work properly when connected together.
- Example: After login, can the user open their dashboard correctly?

3. System Testing

- Tests the complete system as a whole.

- Example: Does the website meet all functional requirements, and is it secure and fast?

4. User Acceptance Testing (UAT)

- The final test, done by the client or real users.
- It checks if the system meets their business needs and is ready to go live.

Why Do We Use the V-Model?

The V-Model is helpful because it fixes the problems of older methods like Waterfall. It focuses on testing early and often.

Advantages:

Early Bug Detection: Problems are found earlier, so fixing them costs less time and money.

- **Better Quality:** Every step in development has a matching test, so fewer mistakes are missed.
- **Clear Process:** It's very well-organized and easy to manage, even in big projects.
- **Good for Critical Projects:** Used in industries like **healthcare, banking, and aviation**, where quality and safety are very important.

🚫 When Not to Use the V-Model

Even though the V-Model is strong, it doesn't work for every type of project.

Disadvantages:

- **Not Flexible:** If your project requirements change often, this model is hard to adjust.
- **No Early Demo:** You don't get to see a working version of the software until the end.
- **Slow for Small Projects:** It takes time and effort to prepare all documents, so it's not ideal for short-term projects.

Summary

The V-Model is a **step-by-step process** where every development stage has a matching testing stage.
It ensures **quality, structure, and clear documentation**, but it's less flexible compared to modern Agile methods.

Many companies today mix the **V-Model** with **Agile practices**, getting the best of both worlds structure + flexibility.

Written by Rashmi Ruwanthika Athukorala

0 followers · 1 following

Edit profile

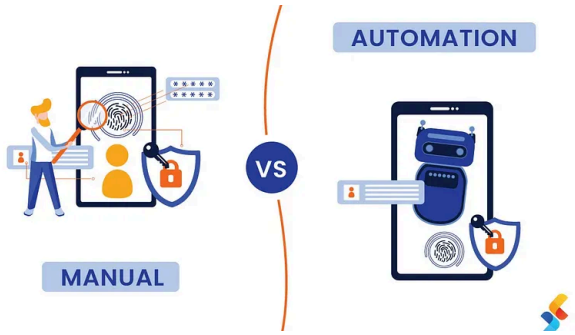
BSc Hons Computer Science and Technology | Undergraduate at Uva Wellassa University, Sri Lanka | QA Enthusiast | Software Tester

No responses yet

Rashmi Ruwanthika Athukorala

What are your thoughts?

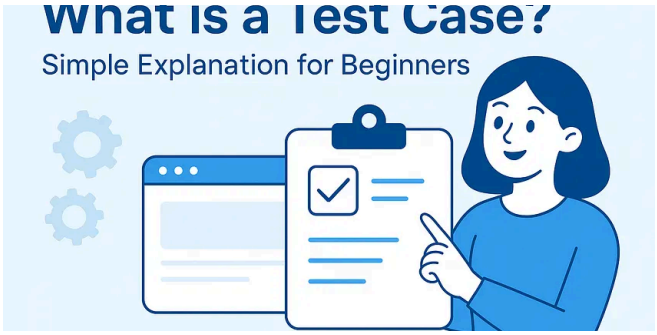
More from Rashmi Ruwanthika Athukorala



Rashmi Ruwanthika Athukorala

🤖 Manual vs. Automated Testing: Which Should You Choose for You...

Choosing the right testing approach is crucial for delivering high quality software. Should...



Rashmi Ruwanthika Athukorala

What Is a Test Case? A Simple and Complete Guide for Beginners

Software today is everywhere in our phones, laptops, cars, banks, hospitals, and even in...