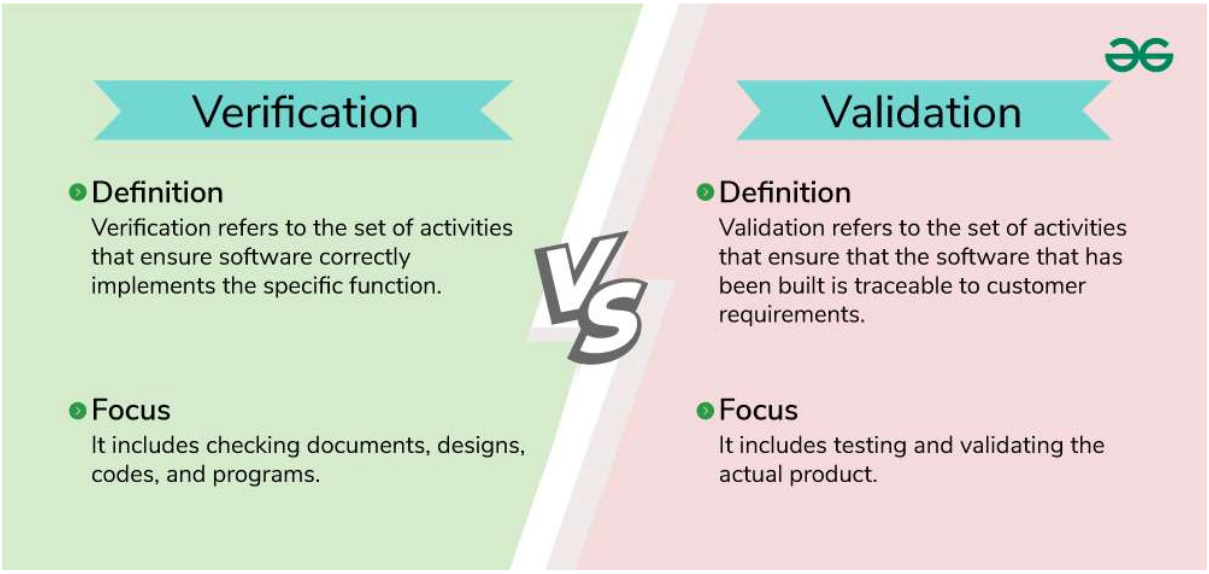
# Verification Vs Validation

Last Updated : 11 Jul, 2025

**Verification and Validation** is the process of investigating whether a software system satisfies specifications and standards and fulfills the required purpose. Verification and Validation both play an important role in developing good software development. Verification helps in examining whether the product is built right according to requirements, while validation helps in examining whether the right product is built to meet user needs.

In this article, we will learning the **Differences between Verification and Validation**.




*Differences between Verification and Validation*

## What is Verification?

**Verification** is the process of checking that software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have. Verification is static testing. Verification means **Are we building the product right?**

## What is Validation?

**Validation** is the process of checking whether the **Software Product** is up to the mark or in other words product has high-level requirements. It is the process of checking the validation of the product i.e. it checks what we are developing is the right product. It is validation of the actual and expected products. Validation is dynamic testing. Validation means **Are we building the right product?**

## Differences between Verification and Validation

Here is the Differences between Verification and Validation

| | Verification | Validation |
|---|---|---|
| **Definition** | Verification refers to the set of activities that ensure software correctly implements the specific function | Validation refers to the set of activities that ensure that the software that has been built is traceable to customer requirements. |
| **Focus** | It includes checking documents, designs, codes, and programs. | It includes testing and validating the actual product. |
| **Type of Testing** | Verification is the **Static testing**. | Validation is **Dynamic testing.** |
| **Execution** | It does *not* include the execution of the code. | It includes the execution of the code. |
| **Methods Used** | Methods used in verification are reviews, walkthroughs, inspections and desk-checking. | Methods used in validation are **Black Box Testing**, **White Box Testing** and **Non-Functional testing**. |
| **Purpose** | It checks whether the software conforms to | It checks whether the software meets the |

| | Verification | Validation |
|---|---|---|
| | specifications or not. | requirements and expectations of a customer or not. |
| **Bug** | It can find the **Bugs** in the early stage of the development. | It can only find the bugs that could not be found by the verification process. |
| **Goal** | The goal of verification is application and software architecture and specification. | The goal of validation is an actual product. |
| **Responsibility** | Verification is typically performed by the quality assurance (QA) team, focusing on reviewing documents, designs, and code to ensure compliance with specified requirements. | Validation is performed by the testing team, which executes the software in real environments to ensure it meets user expectations and requirements. |
| **Timing** | It comes before validation. | It comes after verification. |
| **Human or Computer** | It consists of checking of documents/files and is performed by human. | It consists of execution of program and is performed by computer. |
| **Lifecycle** | After a valid and complete specification the verification starts. | Validation begins as soon as project starts. |
| **Error Focus** | Verification is for prevention of errors. | Validation is for detection of errors. |

| | Verification | Validation |
|---|---|---|
| **Another Terminology** | Verification is also termed as white box testing or static testing as work product goes through reviews. | Validation can be termed as black box testing or dynamic testing as work product is executed. |
| **Performance** | Verification finds about 50 to 60% of the defects. | Validation finds about 20 to 30% of the defects. |
| **Stability** | Verification is based on the opinion of reviewer and may change from person to person. | Validation is based on the fact and is often stable. |

## Real-World Example of Verification vs Validation

- **Verification Example:** Imagine a team is developing a new mobile banking app. During the verification phase, they review the requirements and design documents. They check if all the specified features like fund transfer, account balance check, and transaction history are included and correctly detailed in the design. They also perform peer reviews and inspections to ensure the design aligns with the requirements. This step ensures that the app is being built according to the initial plan and specifications without actually running the app.
- **Validation Example:** In the validation phase, the team starts testing the mobile banking app on actual devices. They check if users can log in, transfer money, and view their transaction history as intended. Testers perform usability tests to ensure the app is user-friendly and functional tests to ensure all features work correctly. They might also involve real users to provide feedback on the app's performance. This phase ensures that the app works as expected and meets user needs in real-world scenarios.

## Advantages of Differentiating Verification and Validation

Differentiating between verification and validation in software testing offers several advantages:

1. **Clear Communication:** It ensures that team members understand which aspects of the software development process are focused on checking requirements (verification) and which are focused on ensuring functionality (validation).
2. **Efficiency:** By clearly defining verification as checking documents and designs without executing code, and validation as testing the actual software for functionality and usability, teams avoid redundant efforts and streamline their testing processes.
3. **Minimized Errors:** It reduces the chances of overlooking critical requirements or functionalities during testing, leading to a more thorough evaluation of the software's capabilities.
4. **Cost Savings:** Optimizing resource allocation and focusing efforts on the right testing activities based on whether they fall under verification or validation helps in managing costs effectively.
5. **Client Satisfaction:** Ensuring that software meets or exceeds client and user expectations by conducting both verification and validation processes rigorously improves overall software quality and user satisfaction.
6. **Process Improvement:** By distinguishing between verification and validation, organizations can refine their testing methodologies, identify areas for improvement, and enhance the overall **Software development lifecycle (SDLC).**

In essence, clear differentiation between verification and validation in **Software testing** contributes to a more structured, efficient, and successful software development process.

## Conclusion

**Verification** is a static process focused on reviewing and analyzing documentation and design without running the code. It verify that the software is being built correctly according to specifications. In other-

side, **validation** is a dynamic process that involves executing the software to check its functionality, usability, and suitability, verifying the right product is built to meet user needs. Both processes are essential for delivering a high-quality software product.



**Next Article**

Top 50+ Software Engineering Interview Questions and Answers [2025]

**Corporate & Communications Address:**
A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

**Registered Address:**
K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305