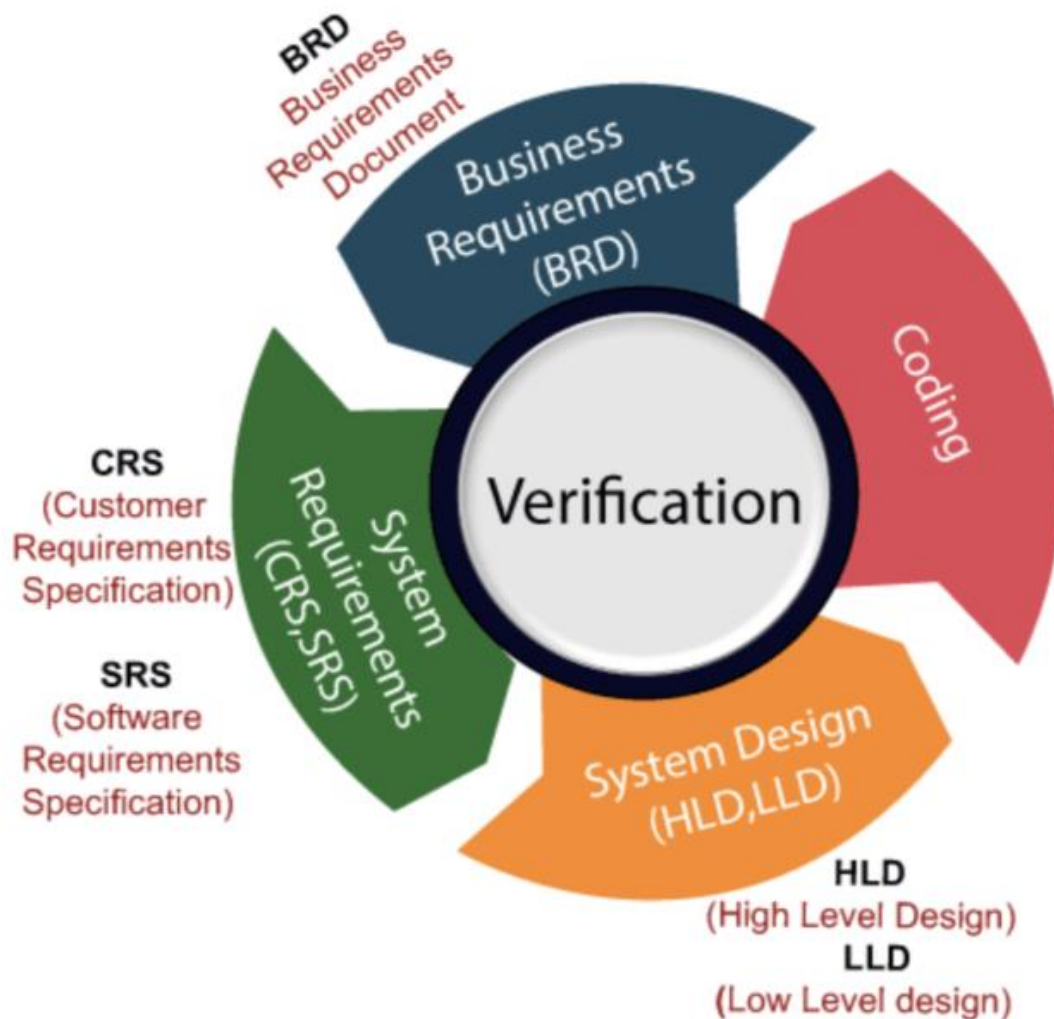


Verification and Validation Testing

What is Verification Testing?

A set of software engineering standards, defined verification testing as “A test of a system to prove that it meets all its specified requirements at a particular stage of its development.”



- **Code reviews, design, and specification analysis** are common components of verification testing.
- Some important documents to be reviewed in this stage are **requirements specification, design blueprints, ER diagrams, database table design**, test cases, test scenarios, etc.
- Verification tests ensure that all development elements (**software, hardware, documentation, and human resources**)

adhere to organizational and team-specific standards and protocols.

- It checks to authenticate that the **system design and architecture** are accurately engineered and error-free.
- Verification checks are often like studying the specifications and checking them against the code logic.

Advantages of Verification Testing

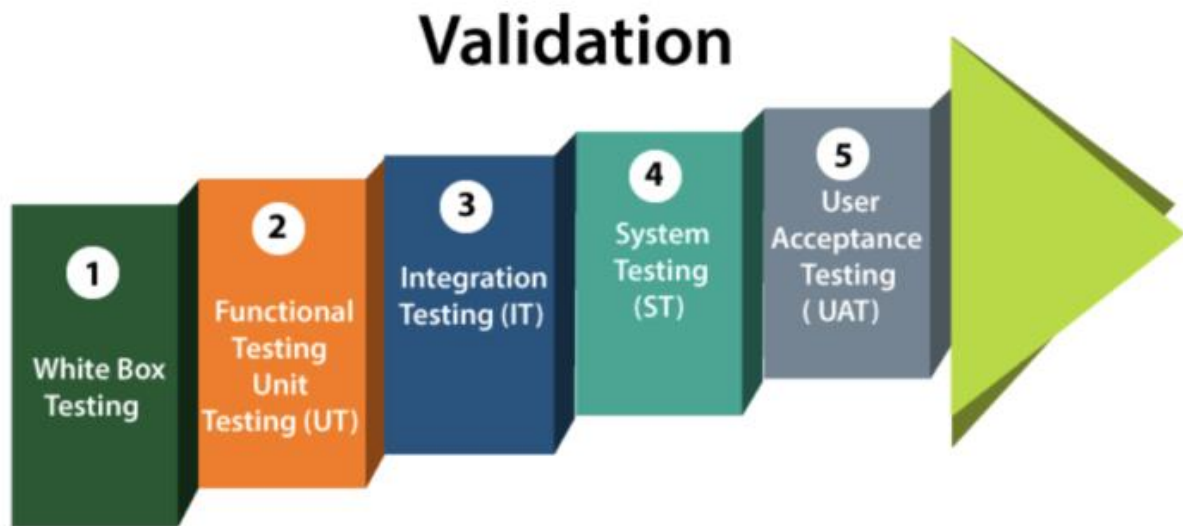
- Early and frequent verification reduces the number of bugs and defects that may show up in later stages.
- By verifying at each stage, developers, product managers, and stakeholders can get more insight into what the product may need to be developed better in the coming stages.
- Verification helps keep software closely aligned with customers and business requirements at every stage. This ensures that developers have to put in less unnecessary work as development continues.

When to use Verification Testing?

- For example, consider a button labelled “**Add to Cart**”. Before creating this button, verification tests would review all relevant requirements previously decided in the ideation phases.
- Let’s say the documentation says the button must be black with the lettering in white. It should be no larger than 10mm X 10mm, and it should constantly be visible in the top right corner of every website product page. Another button with the exact text, colour, and dimensions should be placed under every product on the page.
- Before creating the button, design and requirements documents must be reviewed, and all necessary specifications must be listed before work begins.
- Before working on every feature or element on the page, this must be done so the developers do not miss any guidelines.

What is Validation Testing?

It defines as “An activity that ensures that an end product stakeholder’s true needs and expectations are met.”



Validation testing occurs at the end of a specific module or even after the software has been entirely built. Its primary intent is to ensure the final product matches the stakeholder and customer requirements.

All tests, from unit tests to User Acceptance Tests, are Validation tests.

All validation tests ensure that a system works as planned by running all its functions and tracking tangible, quantifiable results.

Advantages of Validation Testing

- Any bugs missed during verification will be detected while running validation tests.
- If specifications were incorrect and inadequate, validation tests would reveal their inefficacy. Teams will have to spend time and effort fixing them, but it will prevent a bad product from hitting the market.
- Validation tests ensure that the product matches and adheres to customer demands, preferences, and expectations under different conditions (slow connectivity, low battery, etc.)

When to use Validation Testing?

Validation tests must be run after every feature or step in the development process is completed.

- For example, [unit tests](#), a form of validation tests, are run after every unit of code has been created.
- [Integration tests](#) are run after multiple modules have been completed individually and are ready to be combined.
- An essential element of validation testing is the running of [cross browser testing](#). QAs must check that every function, feature, and design element appears and functions as expected on different browser-device-os combinations. For example, does the “Add to Cart” button work perfectly on Google Chrome running Samsung Galaxy A23 and Safari running on iPhone 13?

Difference between Verification and Validation in Software engineering

Verification Testing	Validation Testing
It is the static practice of studying and verifying the specific requirements of a particular stage in development.	It is the dynamic practice of testing the final product after development to ensure it meets customer requirements.
It does not require executing code.	It always requires executing code.
This involves only human verification of required assets.	This involves both human and machine-based checking and approval of software.

It uses document reviews, inspections, product walkthroughs, and desk-checking techniques.	It involves various types of product testing – unit tests, integration tests, regression tests, cross-browser and cross-device testing, etc.
It is meant to detect bugs at the beginning of each development phase.	It is meant to detect all unnoticed bugs at the verification stage.
Its targets are specification documents, application and software architecture design docs, ER diagrams, database table design, test cases and scenarios, traceability matrix, etc.	Its target is the product to be used by the customer after public release.
It is undertaken by both developers and testers to ensure that the software adheres to predetermined standards and exactions.	It is primarily undertaken by experienced Quality Assurance engineers who comb through all features of an application to ensure they work as expected.
It comes before validation testing.	It follows verification testing.
It does not require any devices, platforms, browsers, or operating systems for its execution.	It is best executed by using real browsers, devices, and operating systems.