

**Static Testing vs. Dynamic Testing**

**1. Static Testing (Part of Verification)**

* Performed **without executing** the software.
* Involves reviewing documents, code, and design.
* Helps find errors **early**, reducing the cost of fixing bugs later.

✅ **Examples of Static Testing:**

* Code Reviews
* Requirement Reviews
* Design Reviews
* Using Static Analysis Tools (e.g., SonarQube, Checkstyle)

**2. Dynamic Testing (Part of Validation)**

* Performed **by executing** the software.
* Ensures that the software functions correctly in real-time conditions.
* Helps find functional, performance, and security defects.

✅ **Examples of Dynamic Testing:**

* Functional Testing
* Non-Functional Testing (Performance, Security)
* Regression Testing
* Automated Testing

**Key Differences Between Static & Dynamic Testing**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Static Testing** | **Dynamic Testing** |
| **Type** | Verification | Validation |
| **Execution** | No execution required | Requires execution |
| **Purpose** | Prevent defects early | Detect defects in execution |
| **Cost** | Less expensive (early detection) | More expensive (late-stage bug fixing) |
| **Techniques** | Reviews, Inspections | Functional & Performance Testing |

**Verification and Validation (V&V) in Software Testing**

**1.Verification**

**Are we building the product right?**

* Ensures that the software meets the specified requirements **before execution**.
* Focuses on **preventing defects** during the early stages of development.
* Performed through **reviews, inspections, and walkthroughs**.
* It is a **static** process (does not require running the code).

✅ **Examples of Verification Activities:**

* Requirement Reviews
* Design Reviews
* Code Reviews
* Documentation Reviews

**2. Validation**

**Are we building the right product?**

* Ensures that the software works **correctly after execution**.
* Focuses on **detecting defects** by running the application.
* Ensures the final product meets user needs.
* It is a **dynamic** process (requires running the software).

✅ **Examples of Validation Activities:**

* Unit Testing
* Integration Testing
* System Testing
* User Acceptance Testing (UAT)

**Conclusion**

* **Verification** ensures the software is built **correctly** before execution.
* **Validation** ensures the software works **as expected** after execution.
* **Static Testing** (Verification) finds defects **without execution**.
* **Dynamic Testing** (Validation) finds defects **by running the software**.