UNIT II – LEVELS OF TESTING

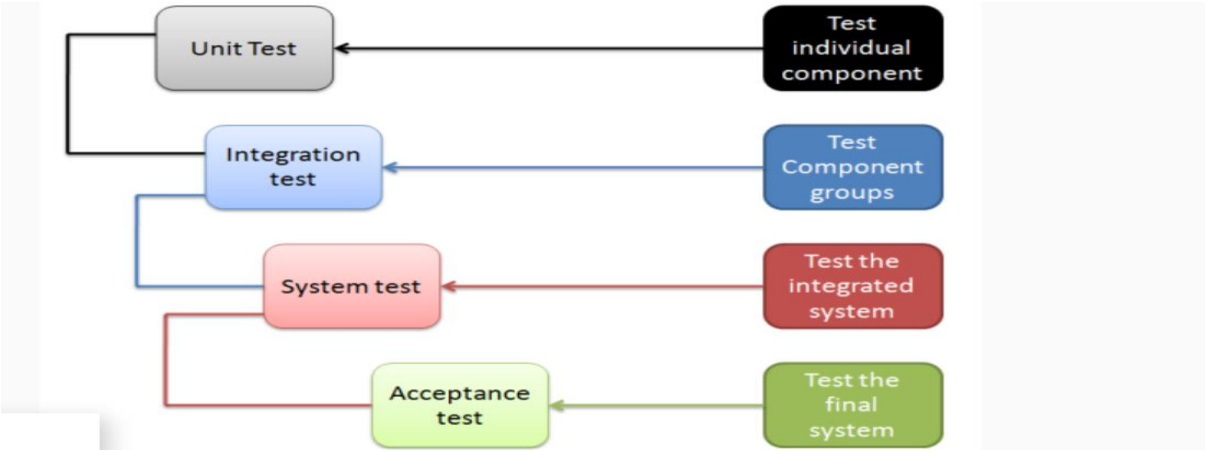
Black Box Testing: Requirements based Testing – Positive and Negative Testing – Boundary Value Analysis – Decision Tables – State Based Testing – Compatibility Testing – User Documentation Testing – Domain Testing. Levels of Testing: System and Acceptance Testing.

Levels of Testing

Software testing is one of the most important phases of the Software Development Life Cycle (SDLC). It ensures that the software meets quality standards by identifying and fixing defects before release. The goal is to deliver reliable software that works correctly for users. To achieve this, the software is tested at different levels.

Software Testing Levels

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing



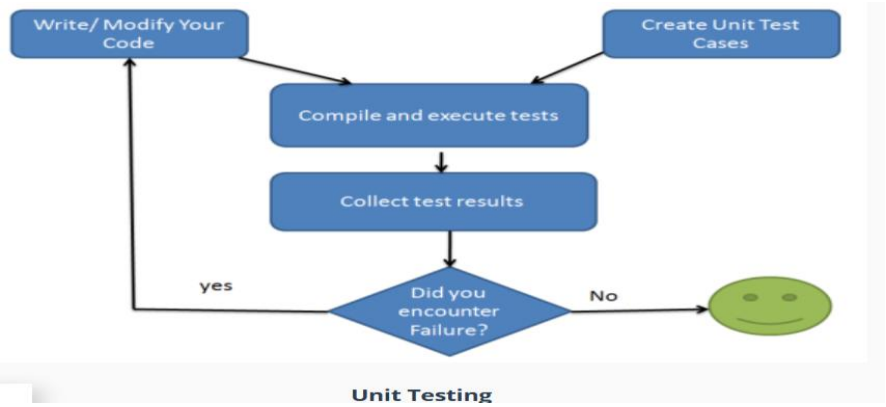
Unit Testing

Unit testing is the process of testing the **smallest testable part (unit)** of a software application. It is the **first level of software testing** and is usually performed by **developers**. Before integration testing begins, all units should be tested individually.

Benefits of Unit Testing:

- Detects defects at an early stage.
- Ensures that each unit works correctly.
- Provides confidence before integration testing.
- Improves code quality and reliability.
- Reduces the time and cost of fixing defects in later stages.

- Creates a strong foundation for the software development process.



Unit Testing

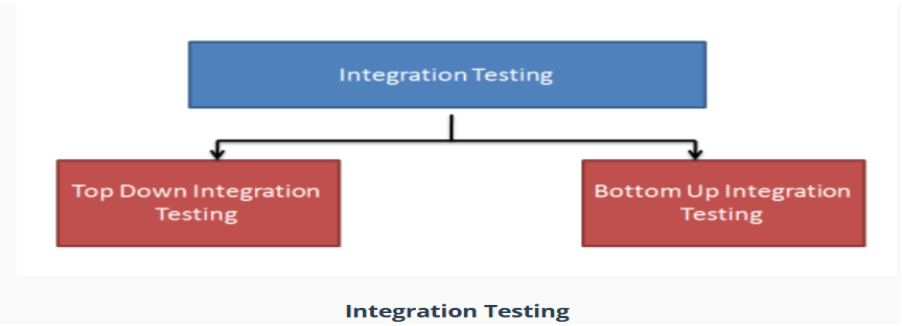
Integration Testing

Integration testing is the **second level of software testing**, performed after unit testing. It checks whether two or more tested modules work correctly when combined. It is usually performed by **software testers** to find defects in the interaction between modules.

Purpose:

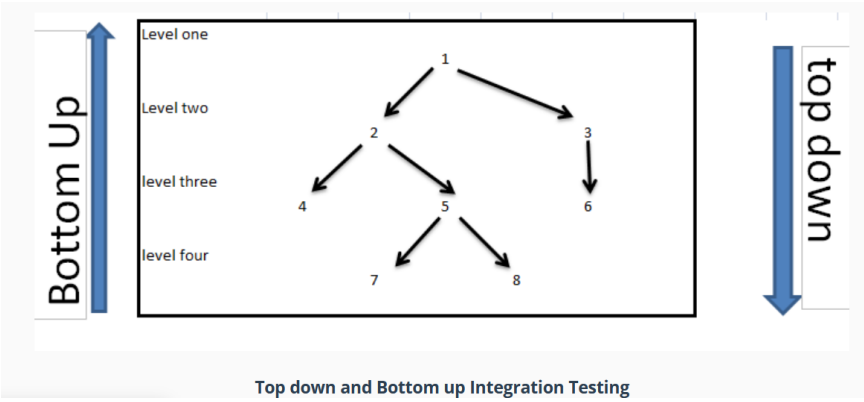
- Verify communication between integrated modules.
- Detect interface, functional, and performance defects.
- Ensure the integrated modules work as expected.

Types of Integration Testing



Integration Testing

1. **Big Bang Integration**
    - All modules are integrated at once and tested together.
    - Suitable for small systems but makes it difficult to identify defects.
2. **Top-Down Integration**
    - Testing starts from the **top-level modules** and gradually adds lower-level modules.
    - **Stubs** (dummy modules) are used when lower modules are not yet available.
3. **Bottom-Up Integration**
    - Testing starts from the **lowest-level modules** and gradually integrates higher-level modules.
    - **Drivers** (dummy programs) are used when higher-level modules are not yet available.



Top down and Bottom up Integration Testing

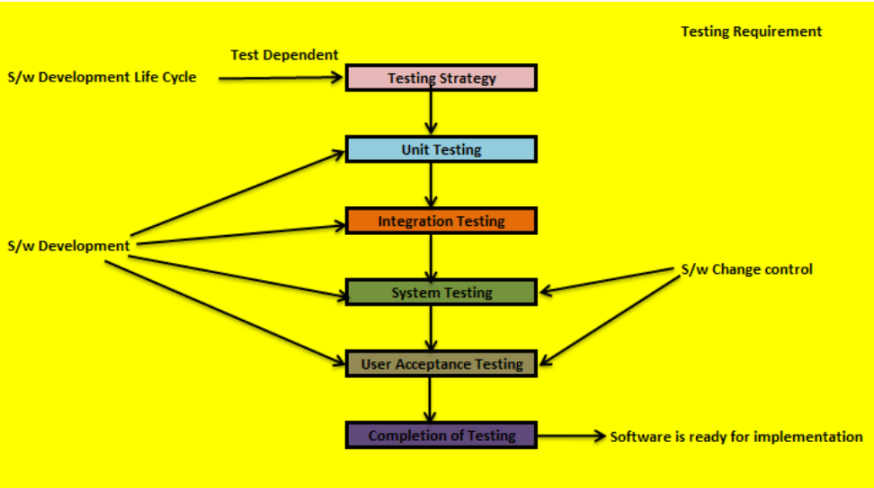
Benefits of Integration Testing:

- Finds defects in module interactions.
- Verifies data flow between modules.
- Ensures all integrated components work together correctly.
- Improves the overall quality and reliability of the software.

System Testing

System testing is the **third level of software testing**, performed after integration testing. In this stage, the **entire software system** is tested as a complete unit to ensure that all integrated components work together correctly. It verifies that the software meets the **functional, technical, and quality requirements** before it is released.

System testing is a **black-box testing** technique, where the application is tested from the **end user's perspective** without examining the internal code. It is usually carried out by a skilled testing team in an environment that closely resembles the real-world deployment environment. This testing checks the software's functionality, performance, reliability, and overall behavior to ensure it satisfies both **business requirements** and **user expectations**.



- Full end-to-end testing of the software.
- Verifying both functional and non-functional requirements.
- Testing the software's behavior in real-world conditions.

Acceptance Testing

Definition:
Acceptance testing is the **final level of software testing**, performed after system testing. Its purpose is to verify that the software satisfies the **customer's requirements**, business needs, and quality standards before it is released. It is carried out using predefined test cases to ensure the software is ready for real-world use and client acceptance.

Types of Acceptance Testing

1. **User Acceptance Testing (UAT)**
    - Performed by the **end users or customers**.
    - Ensures the software meets business requirements and is ready for use.
    - Conducted either at the customer's site or the developer's site.
2. **Operational Acceptance Testing (OAT)**

- Verifies that the system can be **operated and maintained** efficiently.
- Checks backup, recovery, installation, and other operational procedures.

3. **Contract and Regulatory Acceptance Testing**
    - Ensures the software complies with **contractual agreements**, legal requirements, government regulations, and industry standards.
4. **Alpha Testing**
    - Conducted by **test engineers and the development team** before releasing the software to external users.
    - Performed in the developer's environment to identify remaining defects and improve product quality before beta testing.
5. **Beta Testing**
    - Conducted by **actual users** in a real-world environment before the official product release.
    - Helps collect user feedback, identify any remaining issues, and confirm that the software meets customer expectations.

Benefits of Acceptance Testing

- Confirms that the software meets customer requirements.
- Ensures compliance with business, legal, and contractual standards.
- Identifies final defects before release.
- Increases customer confidence and satisfaction.
- Verifies that the software is ready for deployment and real-world use.

