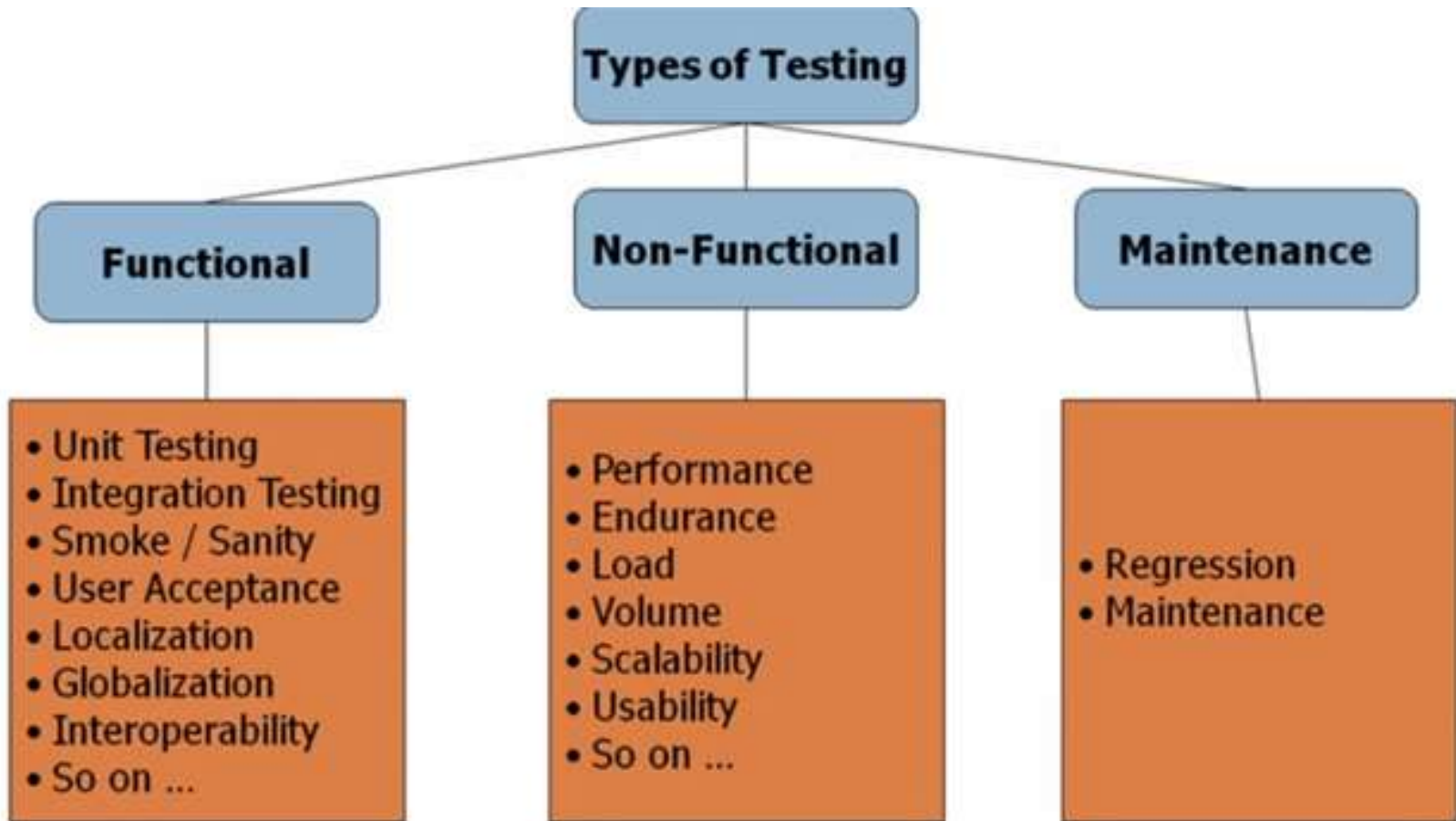


Testing Types

Types of testing



Functional Testing types

- **Unit Testing**
- Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself.
- **It is concerned with functional correctness of the standalone modules.**

Functional Testing types

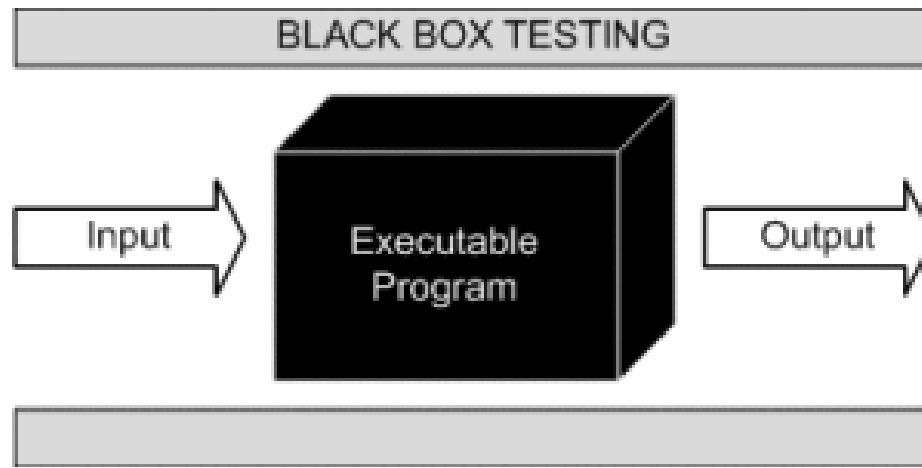
- **Unit Testing - Advantages:**
- Reduces Defects in the Newly developed features or reduces bugs when changing the existing functionality.
- **Reduces Cost of Testing as defects are captured in very early phase.**
- Improves design and allows better refactoring of code.
- Unit Tests, when integrated with build gives the quality of the build as well.

Unit Testing Techniques:

- **Black Box Testing** - Using which the user interface, input and output are tested.
- **White Box Testing** - used to test each one of those functions behavior is tested.
- **Gray Box Testing** - Used to execute tests, risks and assessment methods.

Black Box Testing -

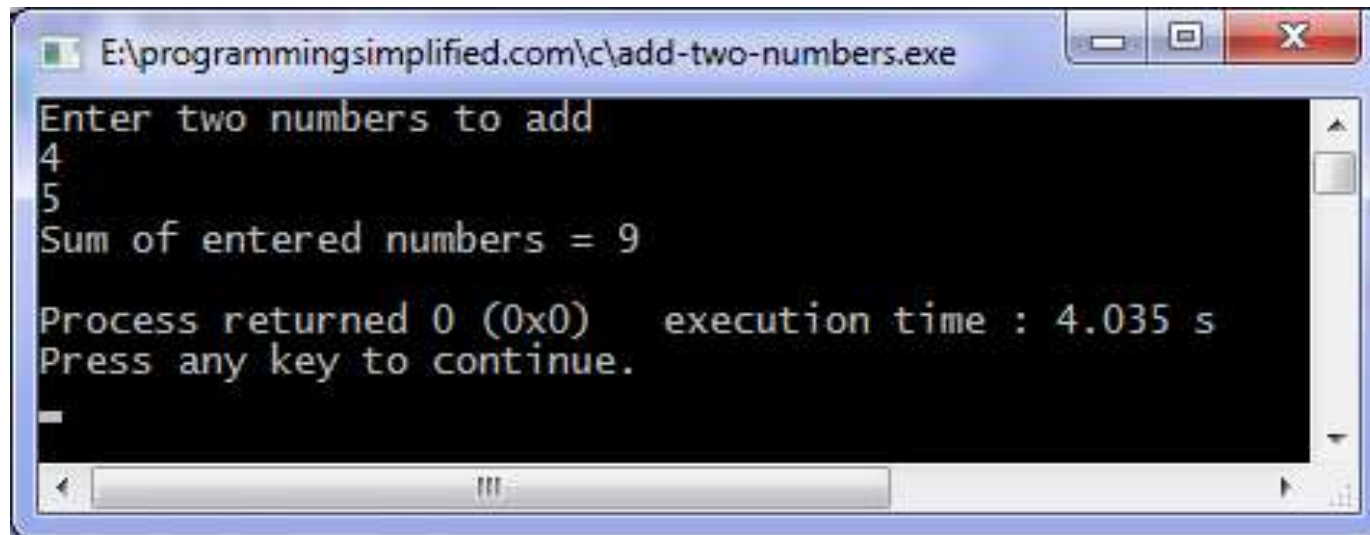
- Also known as Behavioral Testing.
- It is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester.



Black Box Testing -

- We don't look at technical working of an application
- No need of programming skills
- Tests from users perspective
- Input and outputs are checked against expected output

Example



A screenshot of a Windows command prompt window. The title bar shows the file path "E:\programmingsimplified.com\c\add-two-numbers.exe". The command prompt has a black background with white text. It displays the prompt "Enter two numbers to add", followed by the user input "4" and "5" on separate lines. The output is "Sum of entered numbers = 9". Below this, it shows "Process returned 0 (0x0) execution time : 4.035 s" and "Press any key to continue." with a cursor on a new line.

```
E:\programmingsimplified.com\c\add-two-numbers.exe
Enter two numbers to add
4
5
Sum of entered numbers = 9
Process returned 0 (0x0) execution time : 4.035 s
Press any key to continue.
_
```


Black Box Testing Methods

- Graph Based
- Equivalence partitioning
- Boundary value Analysis
- Orthogonal Arrays

White box Testing

- Defined as the testing of a software solution's internal *structure, design, and coding*.
- Also known as clear box testing, Transparent testing and glass box testing.
- **Applied on all levels of SDLC**
- Tests are based on code structure and conditions
- **All logical decisions , loops, and internal data structures of code are verified**

White box testing-techniques

- Basis Path Testing
- Control Structure Testing

Integration testing

- **INTEGRATION TESTING** is a level of software testing where individual units are combined and tested as a group.
- The purpose of this level of testing is to expose faults in the interaction between integrated units.
- **Test drivers and test stubs are used to assist in Integration Testing.**

Integration testing techniques

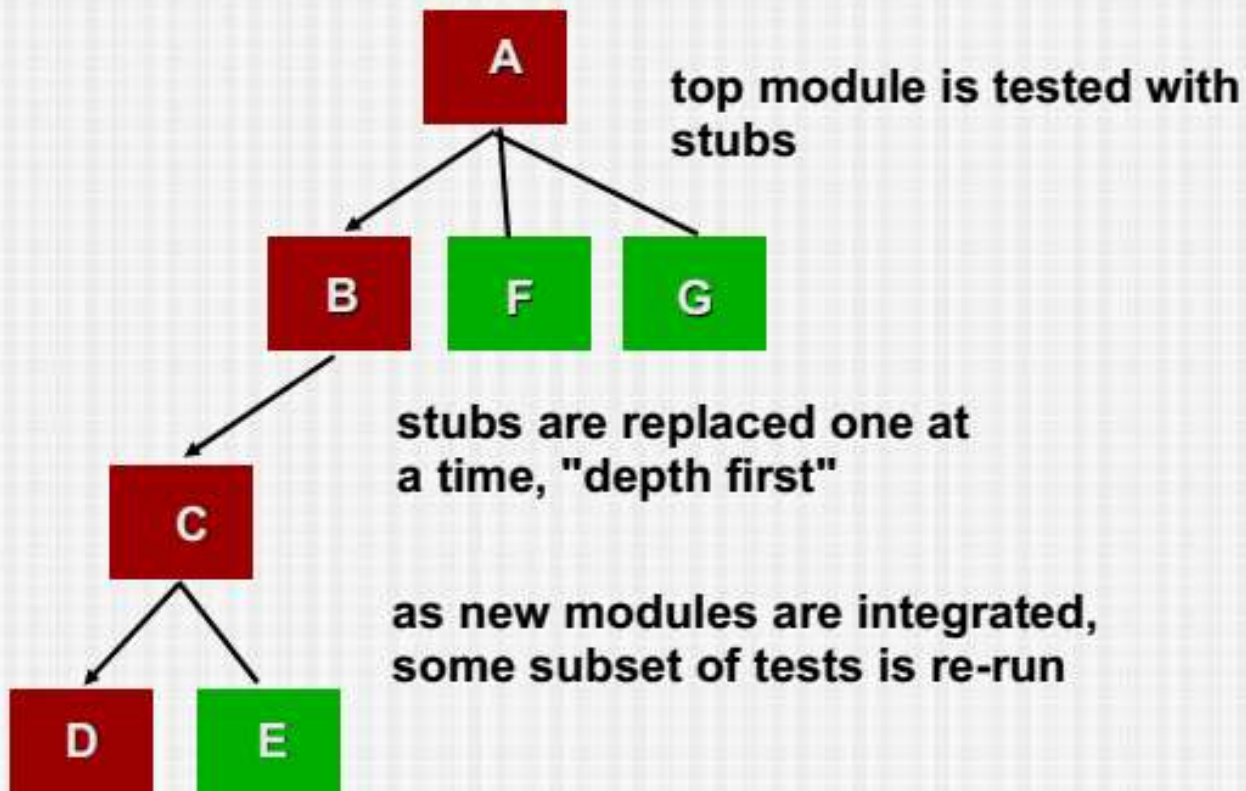
- ***Big Bang*** is an approach to Integration Testing where all or most of the units are combined together and tested at one go.
- This approach is taken when the testing team receives the entire software in a bundle.
- So what is the difference between Big Bang Integration Testing and System Testing? Well, the former tests only the interactions between the units while the latter tests the entire system.

Integration testing techniques

- **Top Down** is an approach to Integration Testing where top-level units are tested first and lower level units are tested step by step after that.
- **Bottom Up** is an approach to Integration Testing where bottom level units are tested first and upper-level units step by step after that. This approach is taken when bottom-up development approach is followed. Test Drivers are needed to simulate higher level units which may not be available during the initial phases.
- **Sandwich/Hybrid** is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.
- **Stubs are used which are temporary modules if a module is not ready for integration testing we replace that module with stubs until it get ready**
- **Stub is a called program and driver is a calling program**

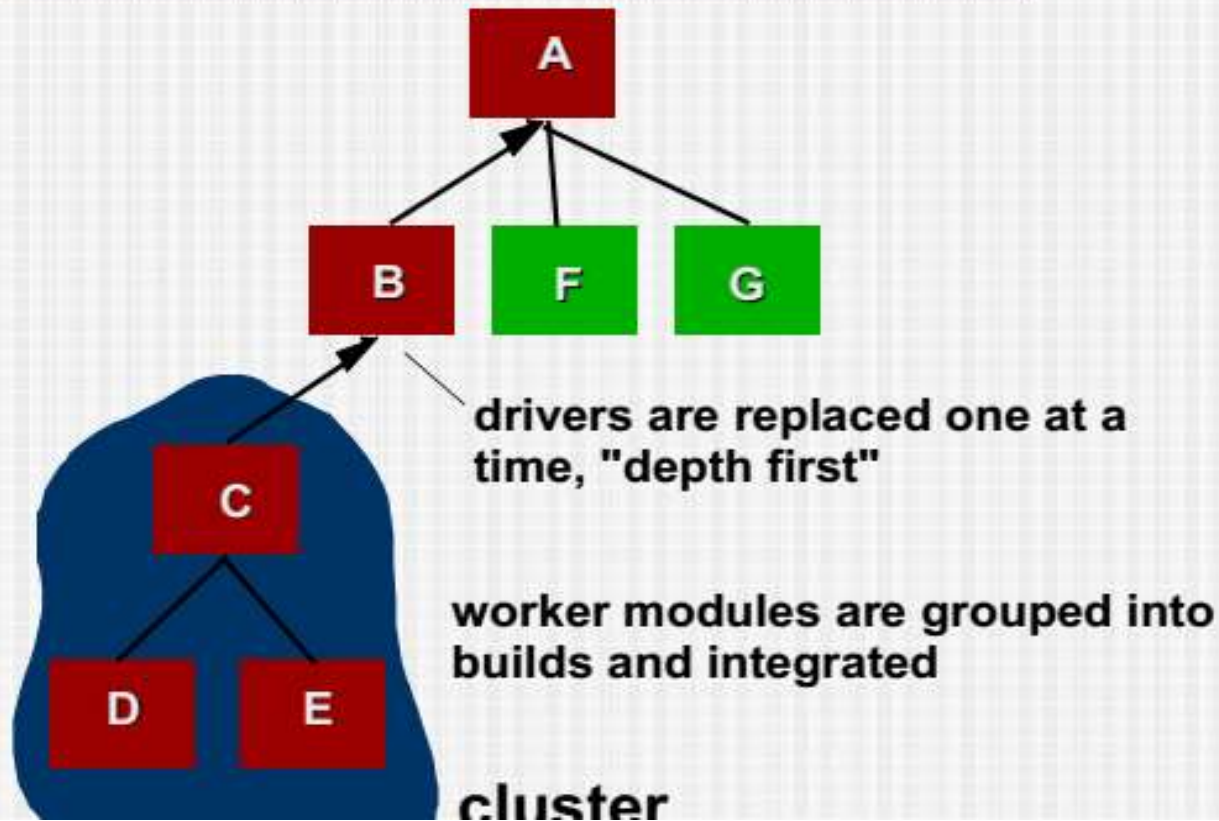
Top-Down Integration

Top Down Integration

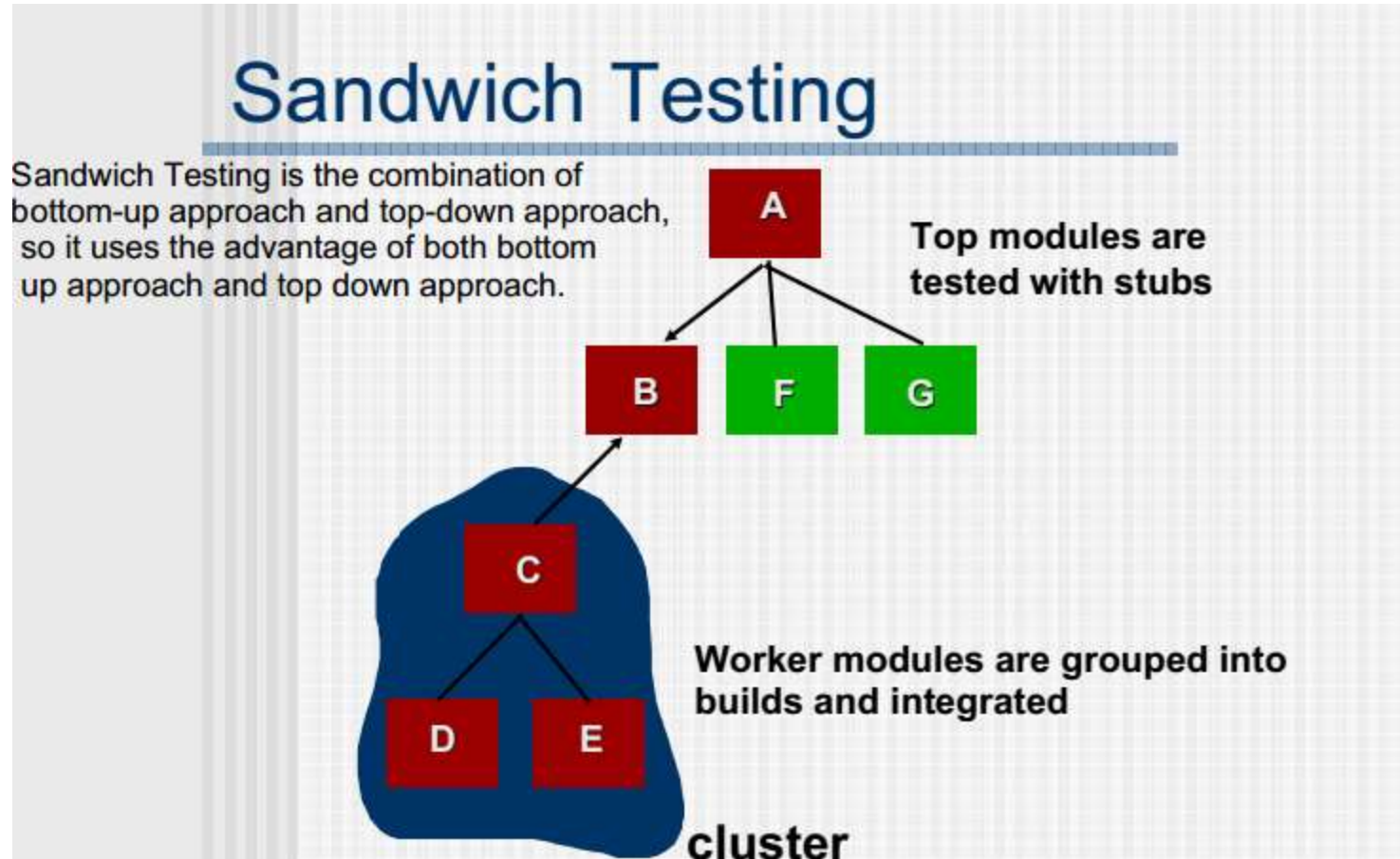


Bottom-Up

Bottom-Up Integration



Sandwich Testing



Grey box testing

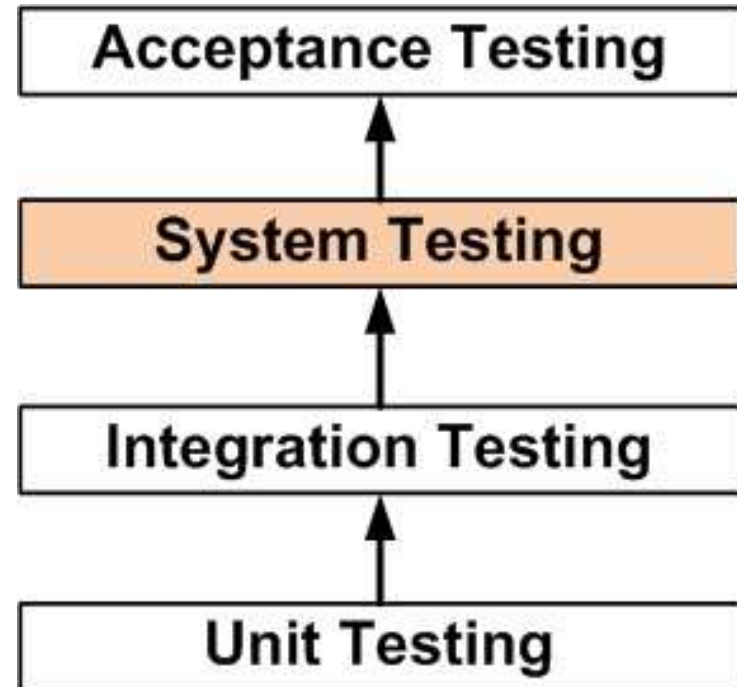
- **GRAY BOX TESTING** is a software testing method which is a combination of Black Box Testing method and White Box Testing method.
- Grey box testing is when the tester has a *partial* understanding of the internal structure in the system under test.
- Grey box testing is a process for debugging software applications by making an input through the front-end, and verifying the data on the back-end.

Gray-box testing Techniques:

- Regression testing
- Pattern Testing
- Orthogonal array testing
- Matrix testing
- Grey-box testing is a perfect fit for Web-based applications.
- Grey-box testing is also a best approach for functional or domain testing.

System testing

- **SYSTEM TESTING** is a level of software testing *where a complete and integrated software is tested*.
- The purpose of this test is to evaluate the system's compliance with the specified requirements.



System testing

- It may include tests based on risks and/or requirements specification, business processes, use cases, or other high level descriptions of system behavior, interactions with the operating system, and system resources.
- System testing is most often the final test on behalf of development to verify that the system to be delivered meets the specification and its purpose may be to find as many defects as possible
- System testing should investigate both **functional** and **non-functional requirements** of the system.

Acceptance testing

- When the development organization has performed its system test and has corrected all or most defects, the system will be delivered to the user or customer for **acceptance testing**.
- Can the system be released?
- Acceptance testing is most often the responsibility of the user or customer
- The goal of acceptance testing is to establish confidence in the system, part of the system or specific non-functional characteristics, e.g. usability, of the system.
- Acceptance testing is most often focused on a validation type of testing, whereby we are trying to determine whether the system is fit for purpose

Alpha & Beta Testing

- Alpha Testing normally takes place in the development environment and is usually done by internal staff.
- The important thing here is that it takes place in the development environment.
- Based on the feedback – collected from the alpha testers – development teams then fix certain issues and improve the usability of the product.
- Beta Testing, also known as “field testing”, takes place in the customer’s environment and involves some extensive testing by a group of customers who use the system in their environment.
- These beta testers then provide feedback, which in turn leads to improvements of the product.

Regression testing

- *Regression testing* is the re-execution of some subset of tests that have already been conducted to ensure that changes have not propagated unintended side effects
- Whenever software is corrected, some aspect of the software configuration (the program, its documentation, or the data that support it) is changed.
- Regression testing helps to ensure that changes (due to testing or for other reasons) do not introduce unintended behavior or additional errors.
- Regression testing may be conducted manually, by re-executing a subset of all test cases or using automated capture/playback tools.