

**SUBJECT: SOFTWARE TESTING AND
QUALITY ASSURANCE**

TOPIC : INTEGRATION TESTING

NAME OF TEACHER: SIMNA V J

ACADEMIC YEAR: 2020-2021

INTEGRATION TESTING

INTEGRATION TESTING

- ❖ **Integration testing** is the process of testing the interface between two software units or module.
- ❖ The purpose of the integration testing is to expose faults in the interaction between integrated units.
- ❖ It's focus on determining the correctness of the interface.



- ◉ Once all the modules have been unit tested, integration testing is performed.
- ◉ The final round of integration involving all components is called final integration testing(FIT) or system integration



- ◉ Integration testing means testing of interfaces.
- ◉ There are 2 types of interfaces .internal and external
- ◉ **Internal:** it provide communication across modules with in a product. it is internal to product and not exposed to customers
- ◉ **External:** They are visible outside product to third party developers



INTEGRATION TEST APPROACHES

Different methodologies for integration testing are

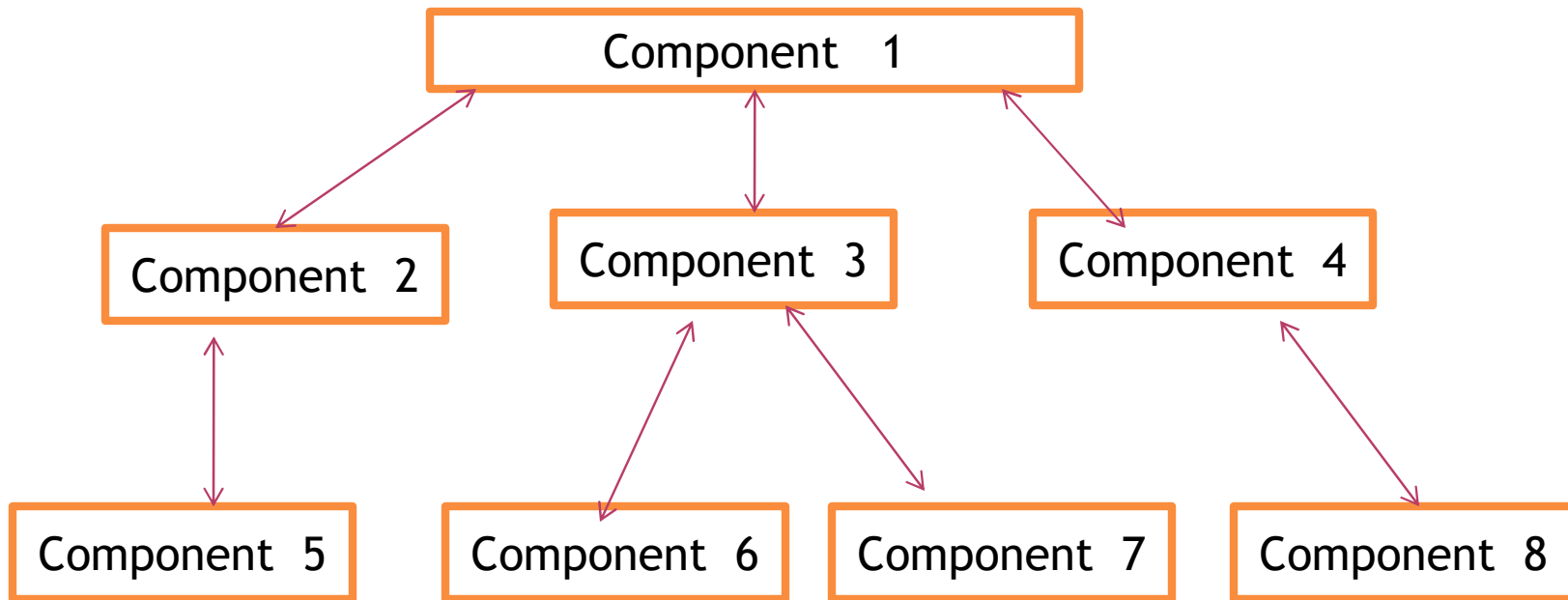
- ◉ Top down integration
- ◉ Bottom up integration
- ◉ Bi directional integration
- ◉ System integration



TOP DOWN INTEGRATION

- ◉ In this integration testing, takes place from top to bottom.
- ◉ First high-level modules are tested and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended.

EXAMPLE



ORDER OF TESTING INTERFACES

step	Interface tested
1	1-2
2	1-3
3	1-4
4	1-2-5
5	1-3-6
6	1-3-6-(3-7)
7	(1-2-5)-(1-3-6-(3-7))
8	1-4-8
9	(1-2-5)-(1-3-6-(3-7))-(1-4-8)



- ◉ If different methods of traversing are used, *breadth first* approach will get you component order such as 1-2,1-3,1-4
- ◉ *Depth first* approach will get you component order such as 1-2-5,1-3-6



- If a set of components and their related interfaces can function without depending on other components or with minimal interface requirement in software, then that set of components are called “*sub system*”



ADVANTAGES

- ⦿ Fault Localization is easier.
- ⦿ Critical Modules are tested on priority. major design flaws could be found and fixed first.

Disadvantages:

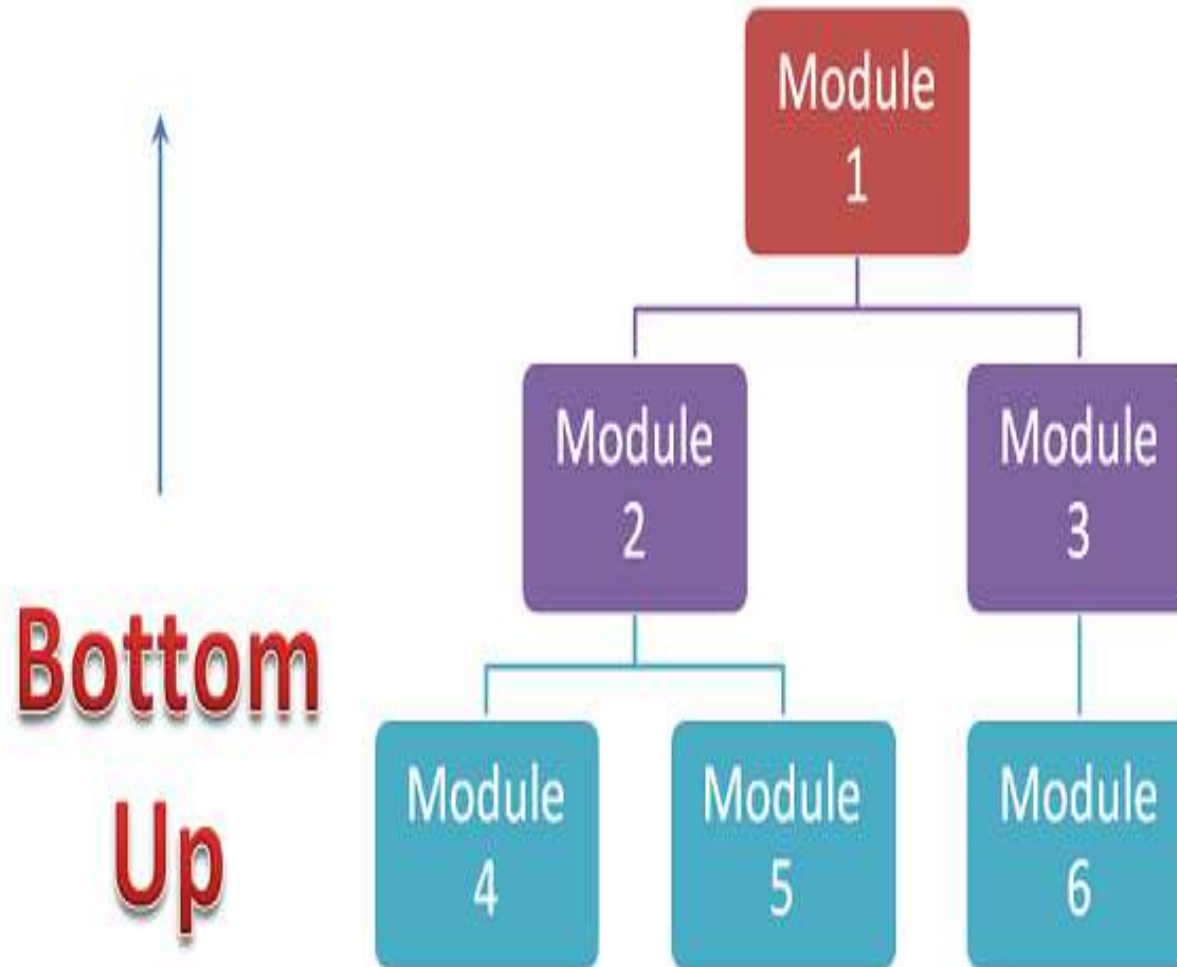
- ⦿ Needs many Stubs.
- ⦿ Modules at a lower level are tested inadequately.



BOTTOM UP INTEGRATION

- ◉ **Bottom-up Integration Testing** is a strategy in which the lower level modules are tested first.
- ◉ These tested modules are then further used to facilitate the testing of higher level modules.
- ◉ The process continues until all modules at top level are tested. Once the lower level modules are tested and integrated, then the next level of modules are formed.

Diagrammatic Representation:



ADVANTAGES

- ◉ Fault localization is easier.
- ◉ No time is wasted waiting for all modules to be developed unlike Big-bang approach



DISADVANTAGES

- ⦿ Critical modules (at the top level of software architecture) which control the flow of application are tested last and may be prone to defects.



BI-DIRECTIONAL INTEGRATION

- **Bi-directional integration** is a strategy in which top level modules are tested with lower level modules at the same time lower modules are integrated with top modules and tested as a system.
- It is a combination of Top-down and Bottom-up approaches therefore it is called **Hybrid Integration Testing**.
- It is also called **sandwich integration**
- It makes use of both stubs as well as drivers.

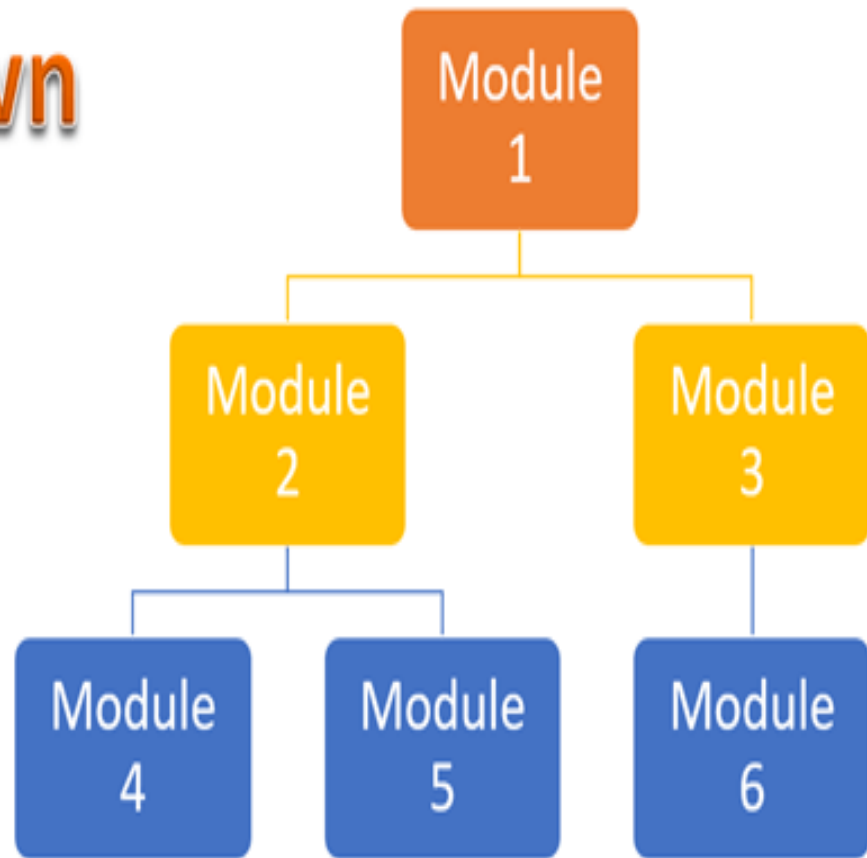
Top Down

↓

Hybrid

↑

Bottom Up



STUBS AND DRIVERS

- ◉ **Stubs and Drivers** are the dummy programs in Integration testing used to facilitate the software testing activity.
- ◉ These programs act as a substitutes for the missing models in the testing.
- ◉ They do not implement the entire programming logic of the software module but they simulate data communication with the calling module while testing.



- ◉ **Stub:** Is called by the Module under Test.
- ◉ **Driver:** Calls the Module to be tested.

- ◉ A **driver** is a function which redirects the requests to some other component and **stubs** simulate the behavior of a missing component



SYSTEM INTEGRATION

- ◉ **System integration(Big Bang Testing)** is an Integration testing approach in which all the components or modules are integrated together at once and then tested as a unit.
- ◉ This combined set of components is considered as an entity while testing.
- ◉ If all of the components in the unit are not completed, the integration process will not execute.

ADVANTAGES

- ◉ Convenient for small systems.



DISADVANTAGES

- ◉ Fault Localization is difficult.
- ◉ Since the Integration testing can commence only after "all" the modules are designed, the testing team will have less time for execution in the testing phase.
- ◉ Since all modules are tested at once, high-risk critical modules are not isolated and tested on priority

