



SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING

(Agile UX/UI)

UNIT 5-Software Testing

Dr.A.SUMITHRA

ASSOCIATE PROFESSOR,

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

SNSCT, Coimbatore.



Software Testing



- Testing is the process of valuating a system with the specific intent of finding errors prior to deliver to end user



STRATEGIC APPROACH OF Software Testing



- The developer should conduct the successful technical reviews to perform the testing successfully . It will eliminate many bugs before testing commences.
- Testing starts with the component level and work from outside towards the integration of the whole computer based system
- Different testing techniques are suitable at different point in time
- Testing is organized by the developer of the software and by an independent test group



STRATEGIC APPROACH OF Software Testing



- Debugging and testing are different activities, then also the debugging should be accommodated in any strategy of testing
- **V&V approach**

Verification : It refers to the set of tasks that ensures that software correctly implements a specific function

Validation : It refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements

Bohem states it in another way :

Verification :”Are we building the product right?”

Validation :”Are we building the right product?”



STRATEGIC APPROACH OF Software Testing



There are many strategies that can be used to test software :

- **Type 1:** testing is done only after the overall system is fully constructed and the errors are found

(this approach simply does not work .It will result in buggy software)

- **Type 2:** You could conduct tests on a daily basis, whenever any part of the system is constructed.

(this approach is effective compared to the previous approach)



STRATEGIC APPROACH OF Software Testing

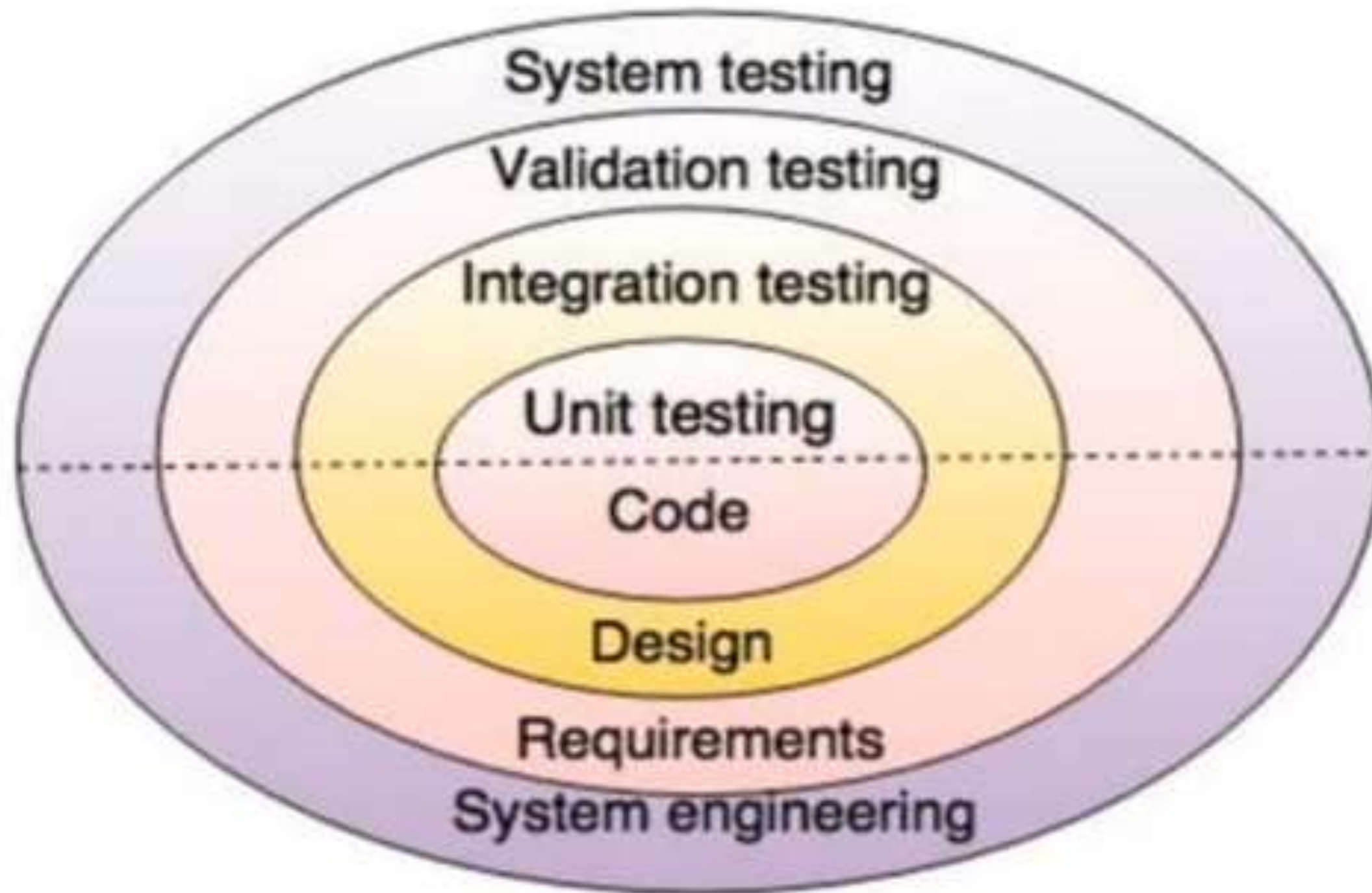


Fig. - Testing Strategy



STRATEGIC APPROACH OF Software Testing



- **Unit Testing** : It starts at the center and each unit is implemented in source code
- **Integration Testing**: It focuses on the construction and design of the software
- **Validation Testing** : Checking all the requirements like functional, behavioral and performance requirement are validate against the construction software
- **System testing**: It confirms all system elements and performance are tested entirely



STRATEGIC APPROACH OF Software Testing



Testing strategy for Procedural point of view:

1. Unit testing
2. Integration testing
3. High order testing
4. Validation testing



STRATEGIC APPROACH OF Software Testing



Testing strategy for Procedural point of view:

The Steps of procedural point of view testing can be shown as :

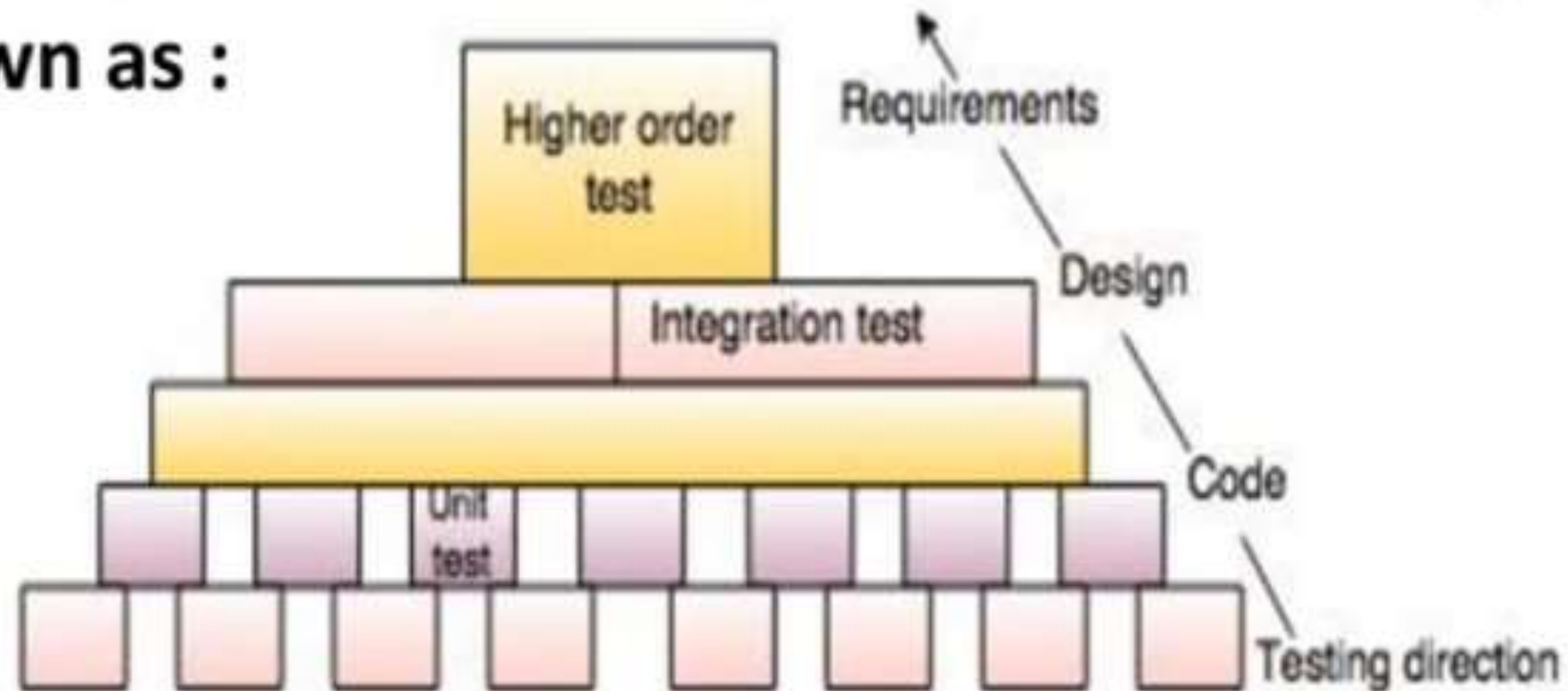


Fig.- Steps of software testing



STRATEGIC APPROACH OF Software Testing

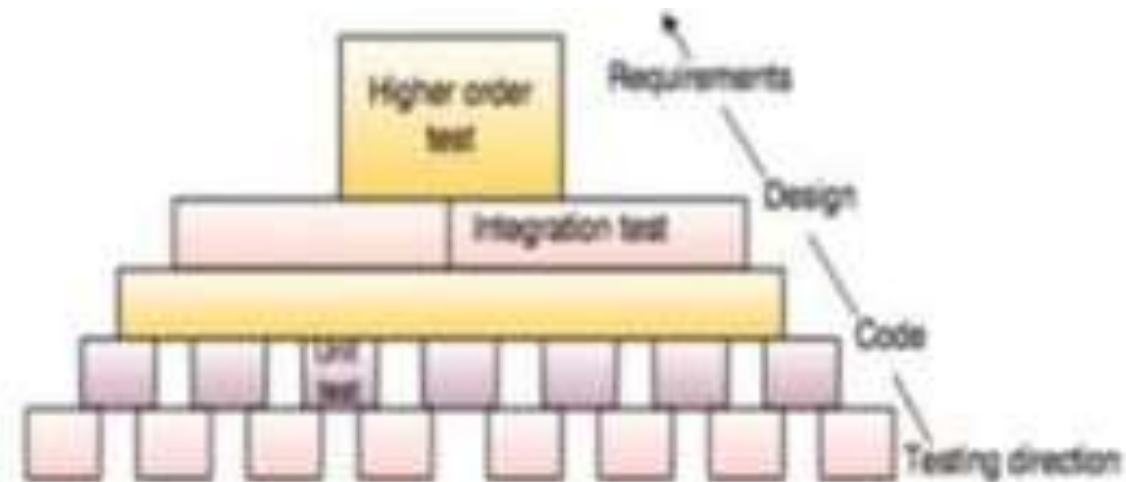


Fig.- Steps of software testing

- Initially, tests focus on each component individually, ensuring that it functions properly as a unit. Hence, the name Unit Testing.
- Unit testing makes heavy use of testing techniques that exercise specific path in a component's control structure to ensure complete coverage and maximum error detection.
- Next, component must be integrated to form the complete software package.



STRATEGIC APPROACH OF Software Testing

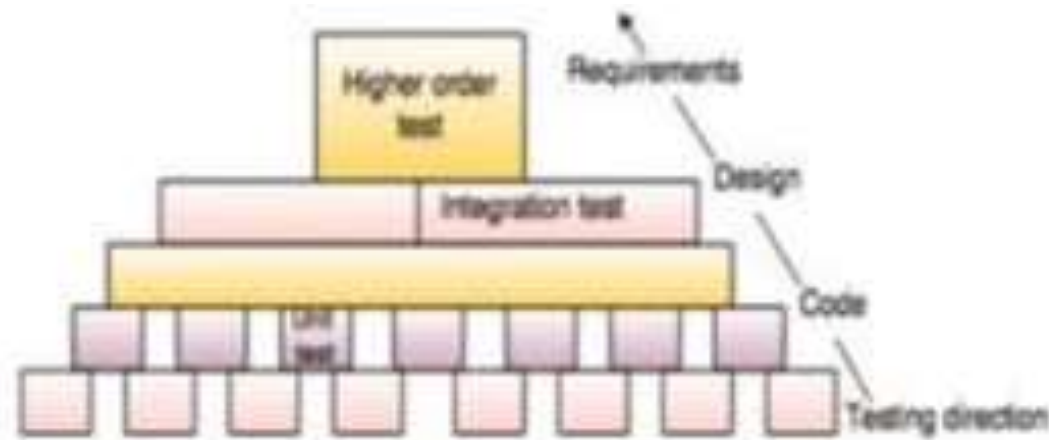


Fig.- Steps of software testing

➤ Integration testing addresses the issues associated with the dual problems of verification and program construction.

➤ After the software has been integrated, a set of high order tests is conducted and validation criteria must be evaluated according to SRS.

➤ Validation testing provides final assurance that the software meets all informational, functional, behavioral, and performance requirements.



STRATEGIC APPROACH OF Software Testing

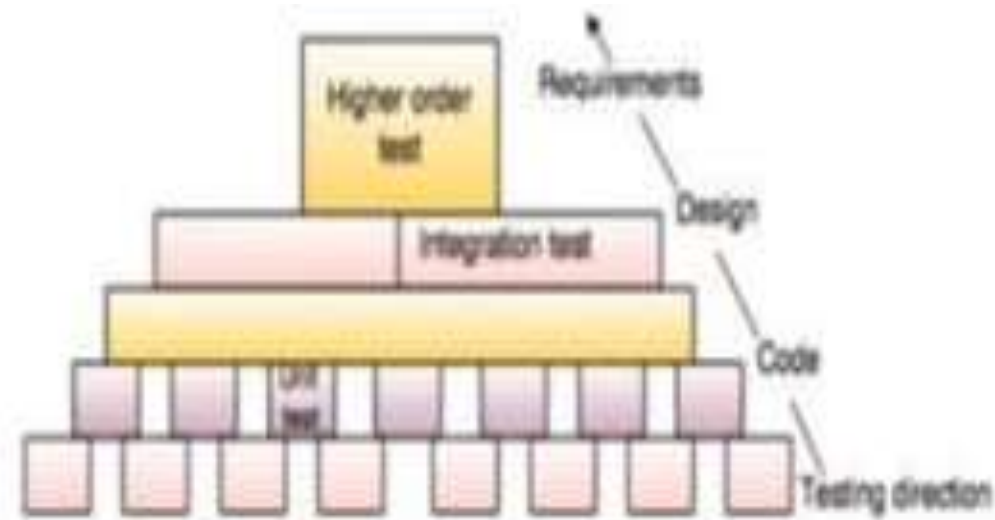


Fig.- Steps of software testing

➤ Software, once validated, must be combined with other system elements like hardware, people, databases, etc.

➤ System testing verifies that all elements mesh properly and that overall system function or performance is achieved.