



CSE 321 Software Engineering

Software Testing

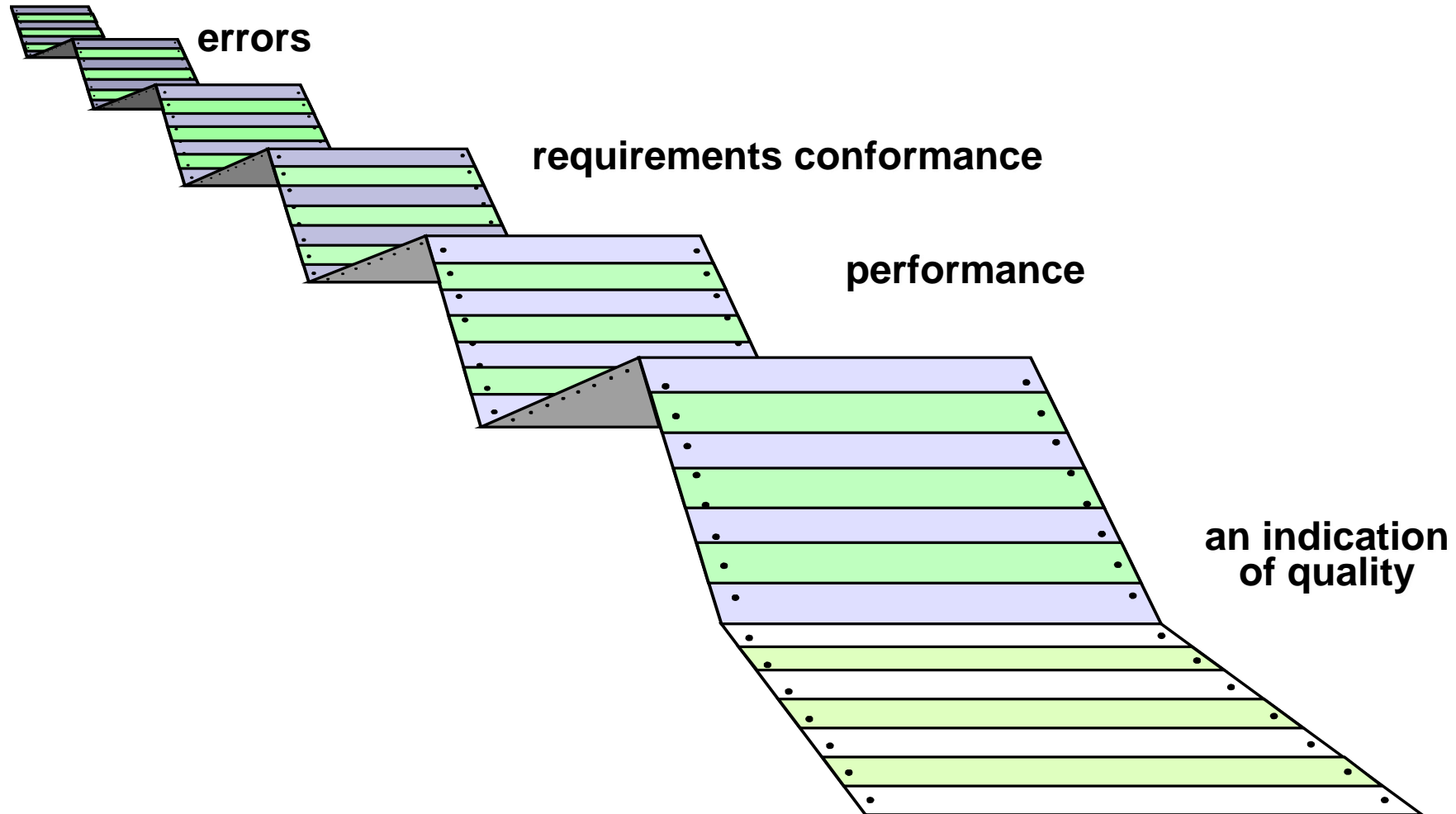
Md. Shamsul Haque

SOFTWARE TESTING



Testing is the process of exercising a program with the specific intent of finding errors prior to delivery to the end user.

WHAT TESTING SHOWS



STRATEGIC APPROACH

- To perform effective testing, you should conduct effective technical reviews. By doing this, many errors will be eliminated before testing commences.
- Testing begins at the component level and works "outward" toward the integration of the entire computer-based system.
- Different testing techniques are appropriate for different software engineering approaches and at different points in time.
- Testing is conducted by the developer of the software and (for large projects) an independent test group.
- Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.

VERIFICATION VS VELIDATION

- *Verification* refers to the set of tasks that ensure that software correctly implements a specific function.
- *Validation* refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Boehm [Boe81] states this another way:

- *Verification*: "Are we building the product right?"
- *Validation*: "Are we building the right product?"

Error, bug, defect, failure



What is a defect?

The variation between the actual results and expected results is known as defect.

If a developer finds an issue and corrects it by himself in the development phase then it's called a defect.

What is a bug?

If testers find any mismatch in the application/system in testing phase then they call it as Bug.

As I mentioned earlier, there is a contradiction in the usage of Bug and Defect. People widely say the bug is an informal name for the defect.

What is an error?

We can't compile or run a program due to coding mistake in a program. If a developer unable to successfully compile or run a program then they call it as an **error**.

What is a failure?

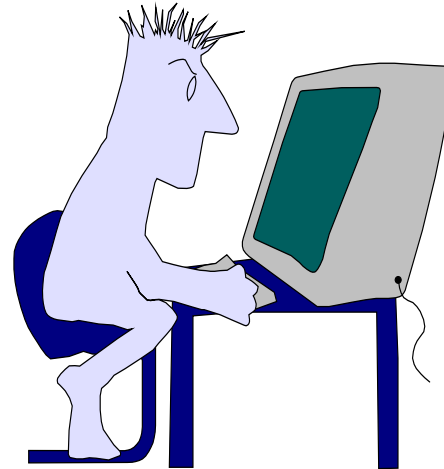
Once the product is deployed and customers find any issues then they call the product as a failure product. After release, if an end user finds an issue then that particular issue is called as **failure**

WHO TEST SOFTWARE



developer

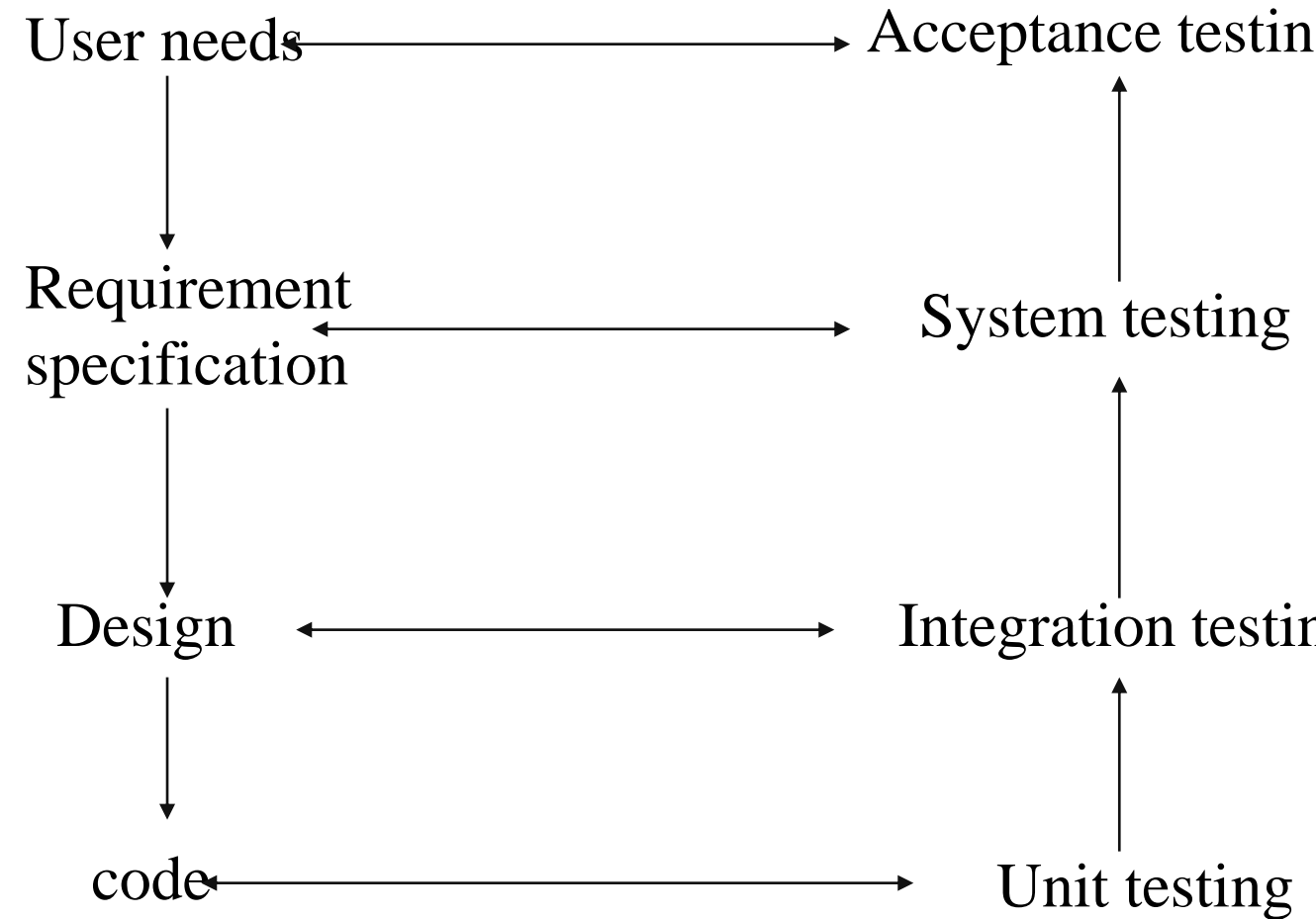
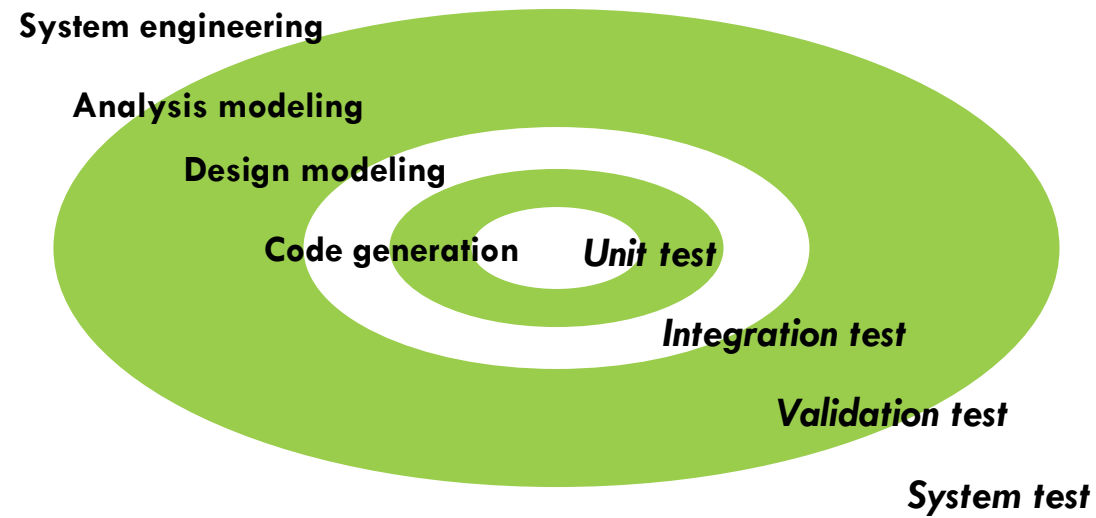
**Understands the system
but, will test "gently"
and, is driven by "delivery"**



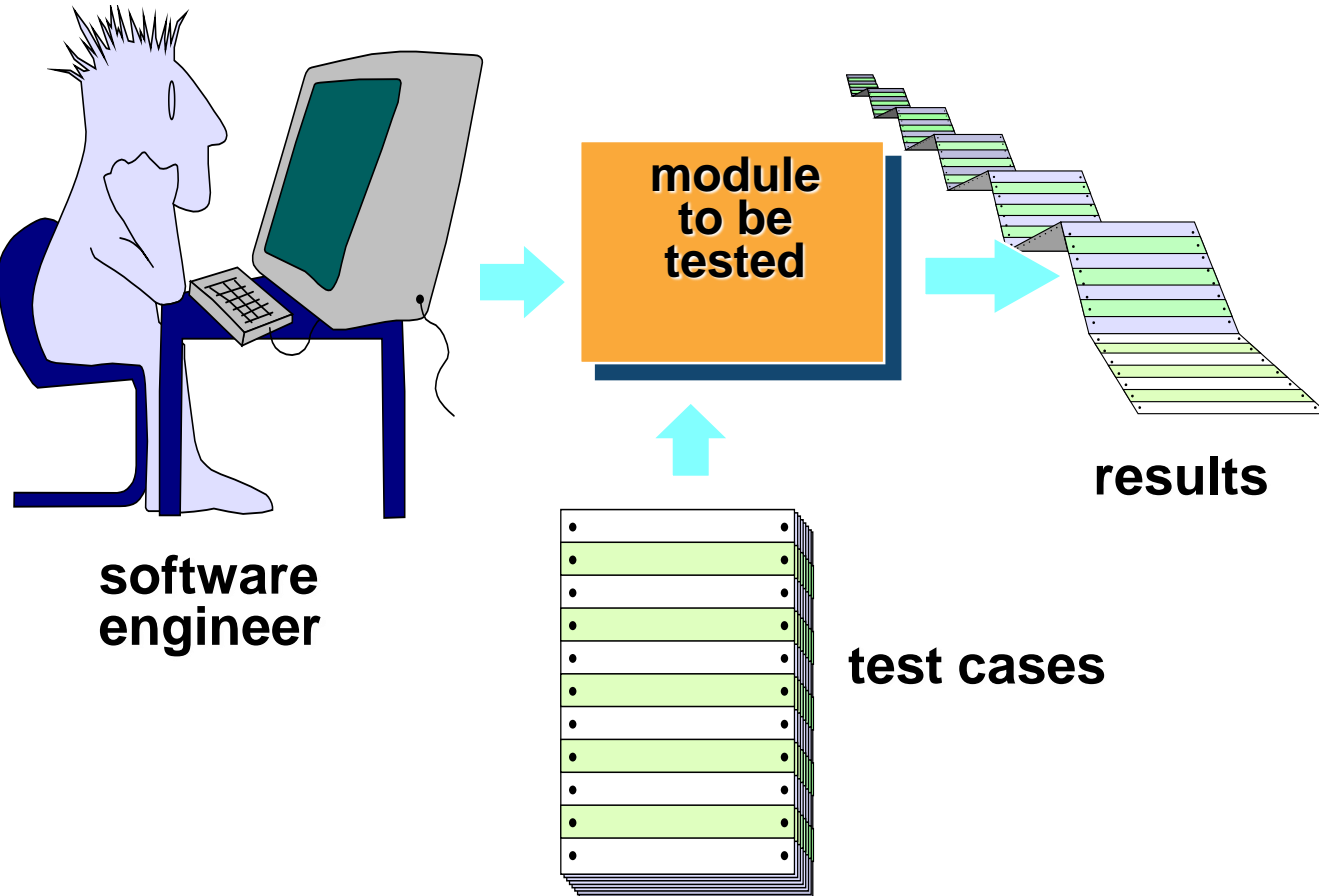
independent tester

**Must learn about the system,
but, will attempt to break it
and, is driven by quality**

TESTING STRATEGY

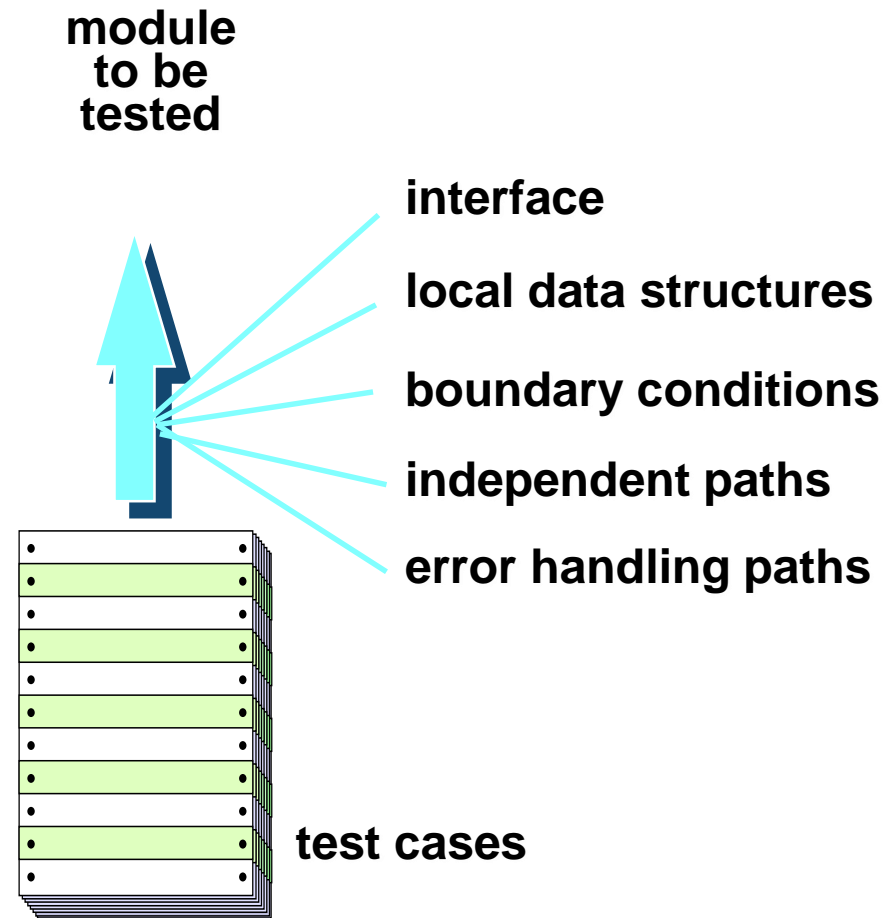


UNIT TESTING



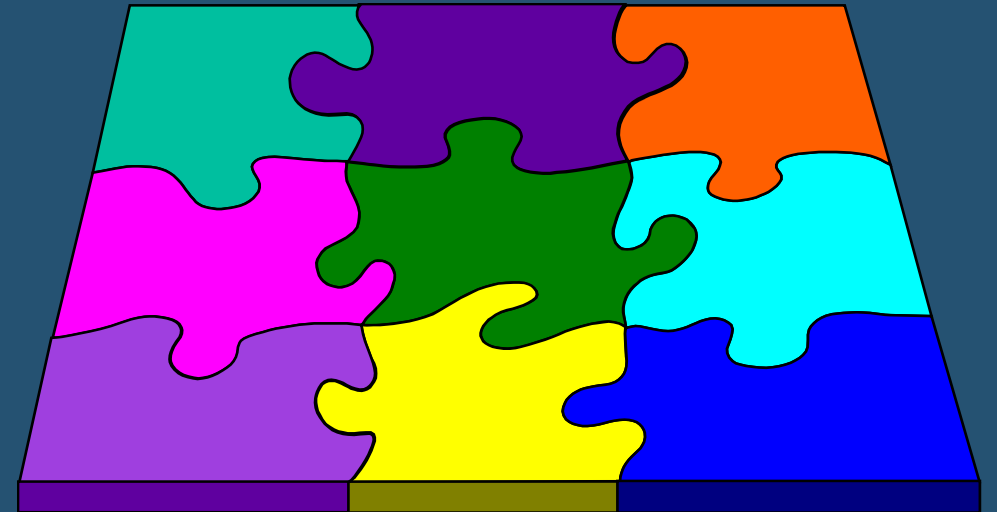
- Different modules tested separately
- Focus: defects injected during coding
- UT is closely associated with coding
- Frequently the programmer does UT;

UNIT TESTING



INTEGRATION TESTING STRATEGIES

- Focuses on interaction of modules in a subsystem
- Unit tested modules combined to form subsystems
- Test cases to “exercise” the interaction of modules in different ways
- May be skipped if the system is not too large



SYSTEM TESTING

- Entire software system is tested
- Focus: does the software implement the requirements?
- Validation exercise for the system with respect to the requirements
- Generally the final testing stage before the software is delivered
- May be done by independent people
- Defects removed by developers
- Most time consuming test phase

ACCEPTANCE TESTING

- Focus: Does the software satisfy user needs?
- Generally done by end users/customer in customer environment, with real data
- Only after successful AT software is deployed
- Any defects found, are removed by developers
- Acceptance test plan is based on the acceptance test criteria in the SRS

OTHER FORM OF TESTING

- Performance testing
 - tools needed to “measure” performance
- Stress testing
 - load the system to peak, load generation tools needed
- Regression testing
 - test that previous functionality works alright
 - important when changes are made
 - Previous test records are needed for comparisons
 - Prioritization of testcases needed when complete test suite cannot be executed for a change

TEST CASE DESIGN

- During test planning, have to design a set of test cases that will detect defects present
- Some criteria needed to guide test case selection
- Two approaches to design test cases
 - functional or black box
 - structural or white box
- Both are complimentary; we discuss a few approaches/criteria for both

BLACK BOX TESTING

- Software tested to be treated as a block box
- Specification for the black box is given
- The expected behavior of the system is used to design test cases
- i.e test cases are determined solely from specification.
- Internal structure of code not used for test case design

BLACK BOX TESTING...

- Premise: Expected behavior is specified.
- Hence just test for specified expected behavior
- How it is implemented is not an issue.
- For modules, specification produced in design specify expected behavior
- For system testing, SRS specifies expected behavior

BLACK BOX TESTING...

- Most thorough functional testing - exhaustive testing
 - Software is designed to work for an input space
 - Test the software with all elements in the input space
- Infeasible - too high a cost
- Need better method for selecting test cases
- Different approaches have been proposed

WHITE BOX TESTING

- Black box testing focuses only on functionality
 - What the program does; not how it is implemented
- White box testing focuses on implementation
 - Aim is to exercise different program structures with the intent of uncovering errors
- Is also called *structural testing*
- Various criteria exist for test case design
- Test cases have to be selected to satisfy coverage criteria

Thank You