Software Engineering

Week # 7

LECTURER: ABDULRAHMAN QAIM

Program testing

Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use.

When you test software, you execute a program using artificial data.

You check the results of the test run for errors, anomalies or information about the program’s non-functional attributes.

Can reveal the presence of errors NOT their absence.

Testing is part of a more general verification and validation process

Program testing goals

To demonstrate to the developer and the customer that the software meets its requirements.

- For custom software, this means that there should be at least one test for every requirement in the requirements document. For generic software products, it means that there should be tests for all of the system features, plus combinations of these features, that will be incorporated in the product release.

To discover situations in which the behavior of the software is incorrect, undesirable or does not conform to its specification.

- Defect testing is concerned with rooting out undesirable system behavior such as system crashes, unwanted interactions with other systems, incorrect computations and data corruption.

Validation and defect testing

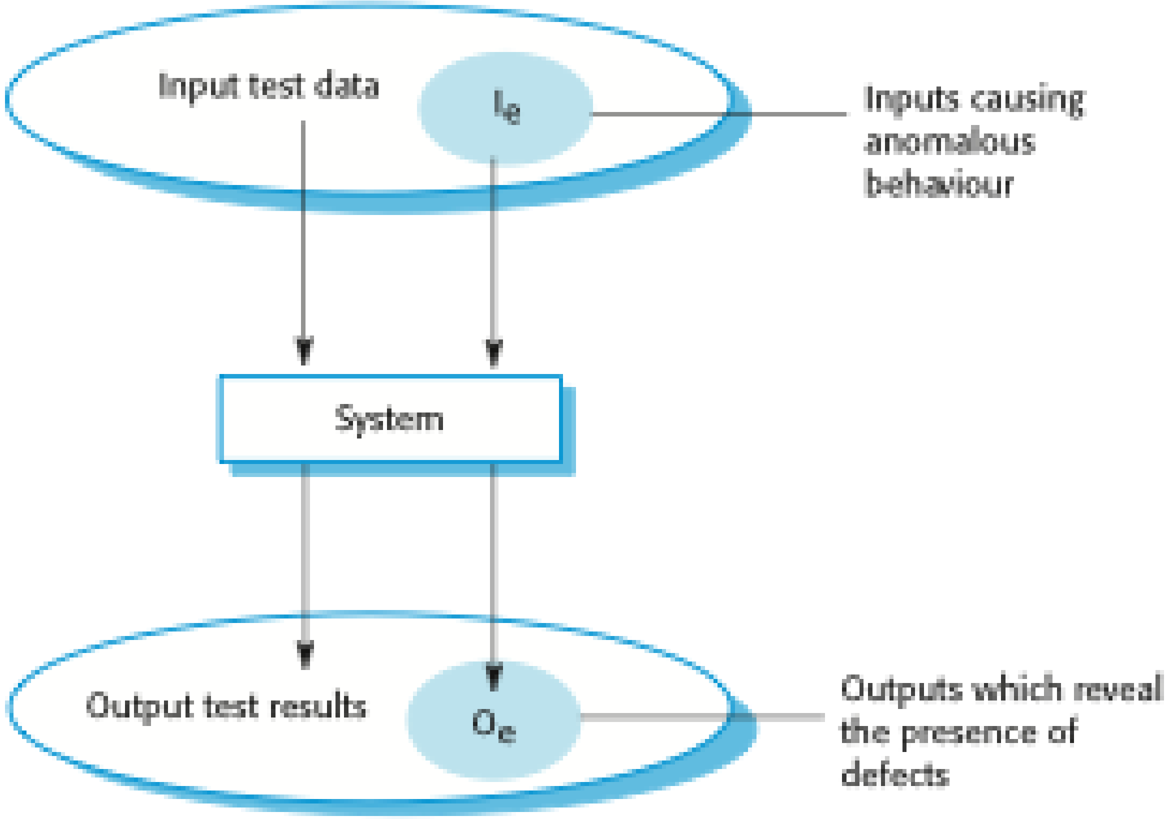
The first goal leads to validation testing

- You expect the system to perform correctly using a given set of test cases that reflect the system's expected use.

The second goal leads to defect testing

- The test cases are designed to expose defects. The test cases in defect testing can be deliberately obscure and need not reflect how the system is normally used.

An input-output model of program testing


Input test data
I_e
Inputs causing anomalous behaviour
System
Output test results
O_e
Outputs which reveal the presence of defects

Verification vs validation

Verification:

"Are we building the product right”.

Validation:

"Are we building the right product”.

V & V confidence

Aim of V & V is to establish confidence that the system is ‘fit for purpose’.

Depends on system’s purpose, user expectations and marketing environment

- Software purpose
  - The level of confidence depends on how critical the software is to an organisation.
- User expectations
  - Users may have low expectations of certain kinds of software.
- Marketing environment
  - Getting a product to market early may be more important than finding defects in the program.

Inspections and testing

Software inspections Concerned with analysis of the static system representation to discover problems (static verification)

- May be supplement by tool-based document and code analysis.

Software testing Concerned with exercising and observing product behaviour (dynamic verification)

- The system is executed with test data and its operational behaviour is observed.

Software inspections

These involve people examining the source representation with the aim of discovering anomalies and defects.

Inspections not require execution of a system so may be used before implementation.

They may be applied to any representation of the system (requirements, design,configuration data, test data, etc.).

They have been shown to be an effective technique for discovering program errors.

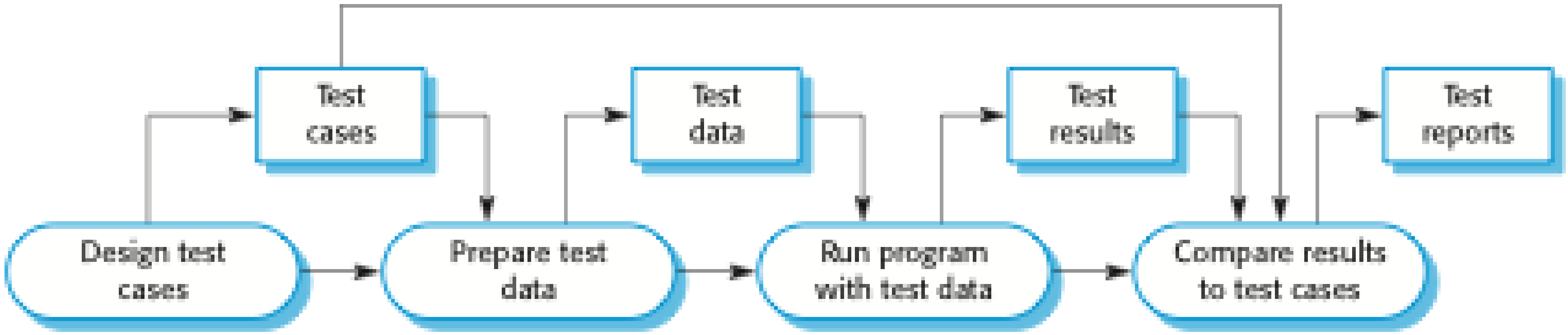
Advantages of inspections

During testing, errors can mask (hide) other errors. Because inspection is a static process, you don’t have to be concerned with interactions between errors.

Incomplete versions of a system can be inspected without additional costs. If a program is incomplete, then you need to develop specialized test harnesses to test the parts that are available.

As well as searching for program defects, an inspection can also consider broader quality attributes of a program, such as compliance with standards, portability and maintainability.

A model of the software testing process



Stages of testing

Development testing, where the system is tested during development to discover bugs and defects.

Release testing, where a separate testing team test a complete version of the system before it is released to users.

User testing, where users or potential users of a system test the system in their own environment.

Development testing

Development testing includes all testing activities that are carried out by the team developing the system.

- Unit testing, where individual program units or object classes are tested. Unit testing should focus on testing the functionality of objects or methods.
- Component testing, where several individual units are integrated to create composite components. Component testing should focus on testing component interfaces.
- System testing, where some or all of the components in a system are integrated and the system is tested as a whole. System testing should focus on testing component interactions.

Unit testing

Unit testing is the process of testing individual components in isolation.

It is a defect testing process.

Units may be:

- Individual functions or methods within an object
- Object classes with several attributes and methods

Object class testing

Complete test coverage of a class involves

- Testing all operations associated with an object
- Setting and interrogating all object attributes
- Exercising the object in all possible states.

Inheritance makes it more difficult to design object class tests as the information to be tested is not localised.