

SOFTWARE TESTING

Static and Performance Testing

Dr. ARIVUSELVAN.K

Associate Professor – (SCORE)

VIT University

Inspection process

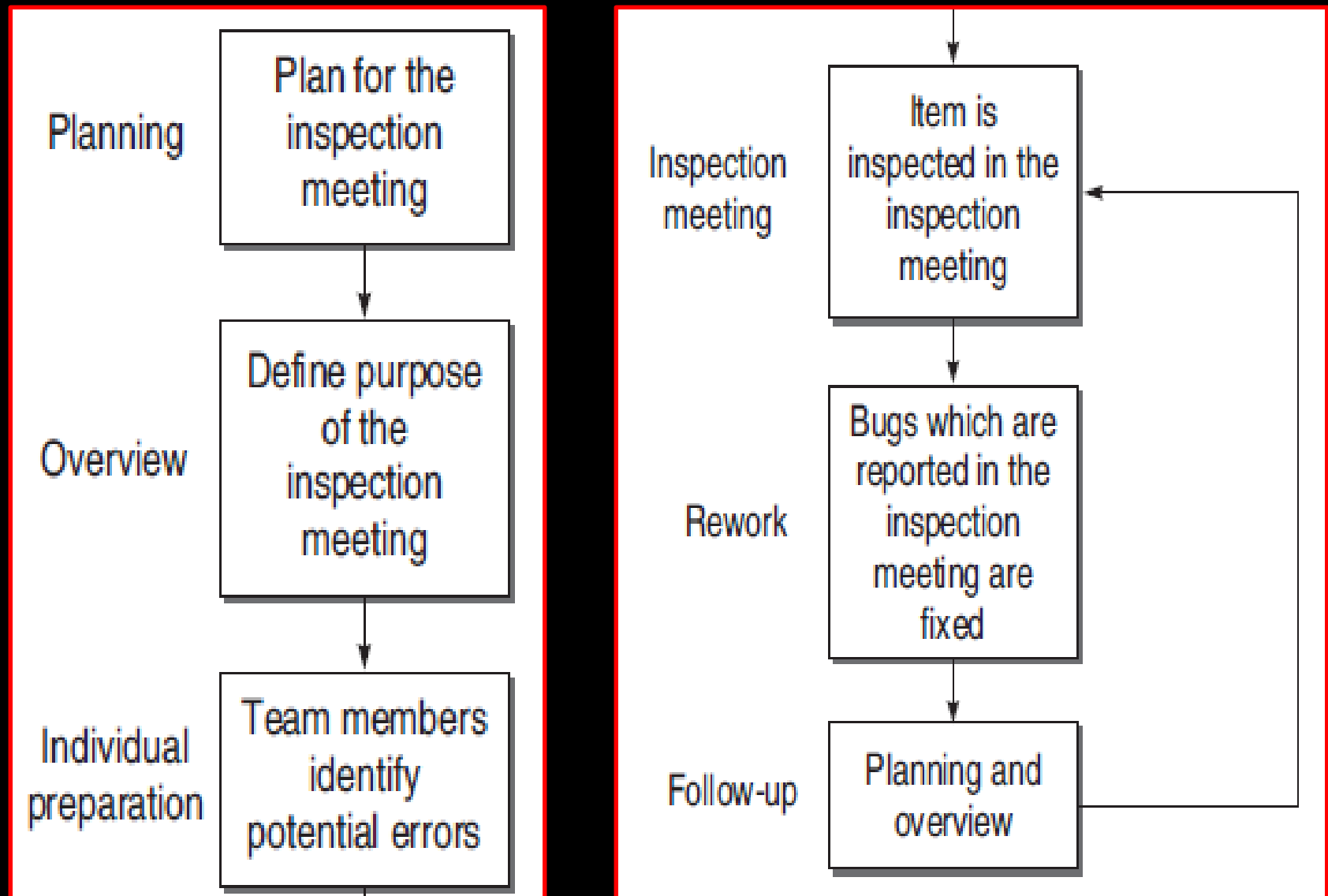
Software inspections were first introduced at **IBM** by **Fagan** in the early **1970s**.

Inspection process is an **in-process manual examination** of an item to **detect bugs**.

An inspection process involves the **interaction of the following elements**:

- **Inspection steps**
- **Role** for participants
- **Item** being inspected

General Inspection process



Variants of inspection process

After **Fagan's** original formal inspection concept, many **researchers** proposed modifications in it..

Table below **lists** some of the **variants** of the **formal inspection**.

Active Design
Reviews (ADRs)

Formal Technical
Asynchronous review
method (FTArm)

Gilb Inspection

Humphrey's
Inspection Process

N-Fold inspections

Phased Inspection

Structured
Walkthrough

Variants of inspection process

Active Design Reviews (ADRs)

Several reviews are conducted targeting a particular type of bugs and conducted by the reviewers who are experts in that area.

Brief **overview** of the **module** being reviewed is presented.

Reviewers are assigned, **sections of the document** to be reviewed and **questionnaires** based on the **bug type**.

The **designers** read the **completed questionnaires** and meet the **reviewers** to **resolve any queries**.



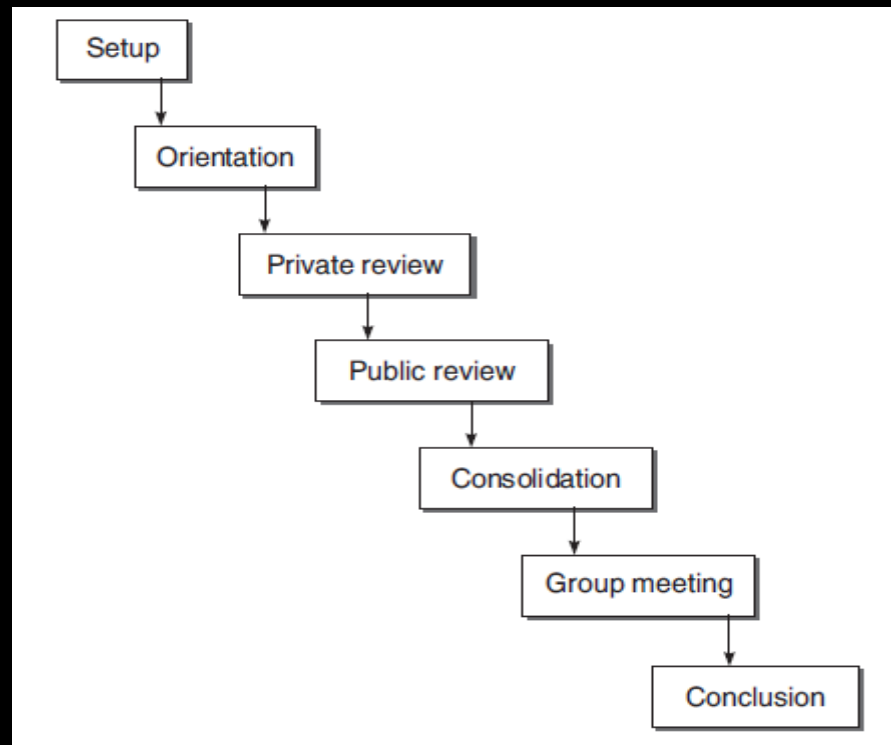
Active design review process

**Formal Technical
Asynchronous review
method (FTArm)**

Inspection process is carried out without really having a meeting of the members. This is a type of asynchronous inspection in which the inspectors never have to simultaneously meet.

In this process, the **meeting phase** of inspection is considered **expensive**.

online version of the **document** is made available to **every member** where they can **add their comments** and **point out the bugs**.



Gilb Inspection

Defect detection is carried out by individual inspector at his level rather than in a group.

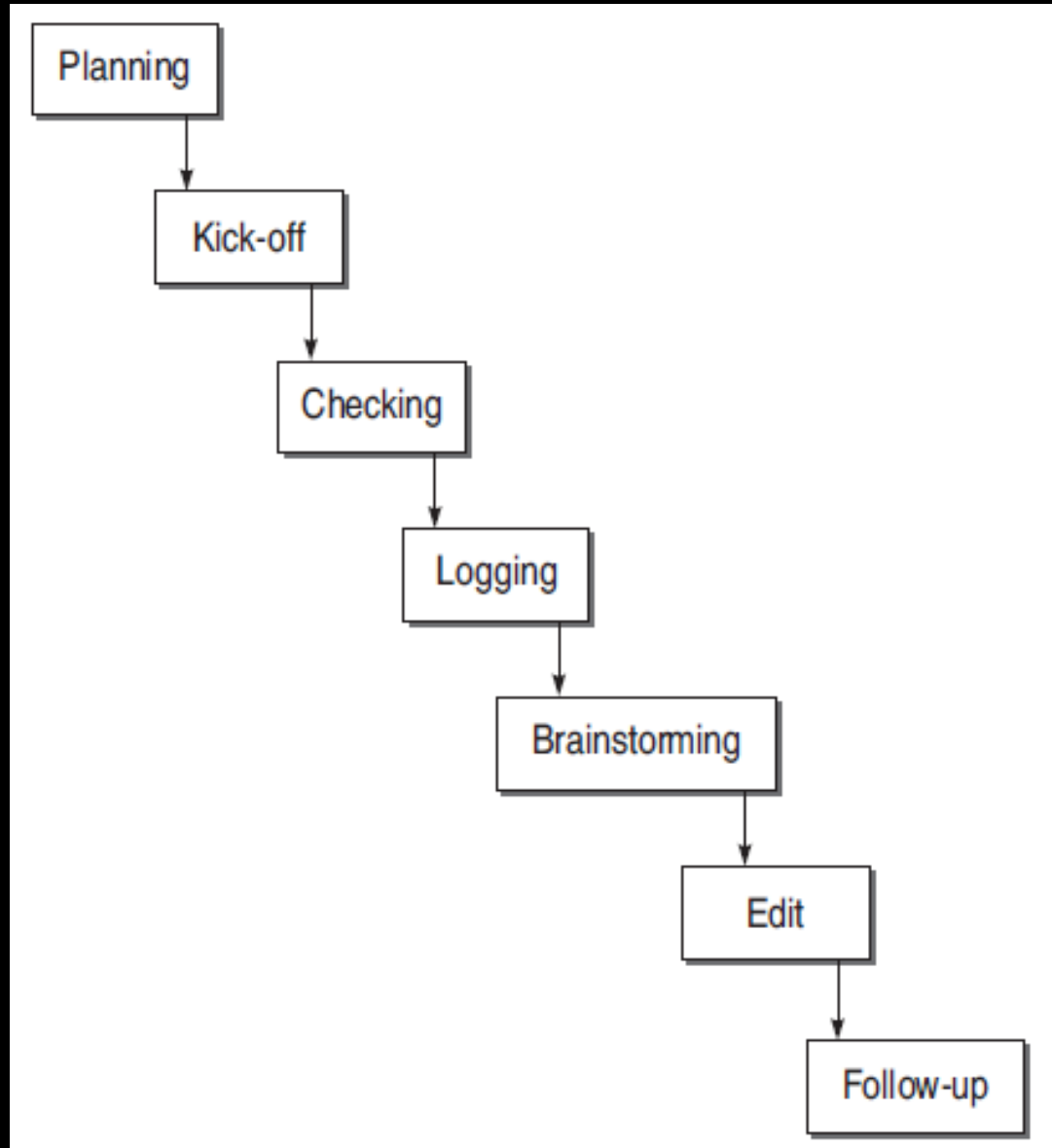
Three different roles are defined in this type of inspection:

Leader, is responsible for **planning** and **running** the inspection.

Author, of the document.

Checker, is responsible for **finding** and **reporting** the defects in the document.

The **inspection process** consists of the following steps:



Planning, The leader determines the **inspection participants** and **schedules** the meeting.

Kick-off, The relevant **documents** are **distributed**, **participants** are assigned **roles** and briefed about the **agenda** of the meeting.

Checking, Each checker **works individually** and finds defects.

Logging, Potential **defects** are collected and logged.

Brainstorm, In this stage, process **improvement suggestions** are **recorded** based on the reported bugs.

Edit, After all the **defects** have been reported, the **author** takes the list and works accordingly.

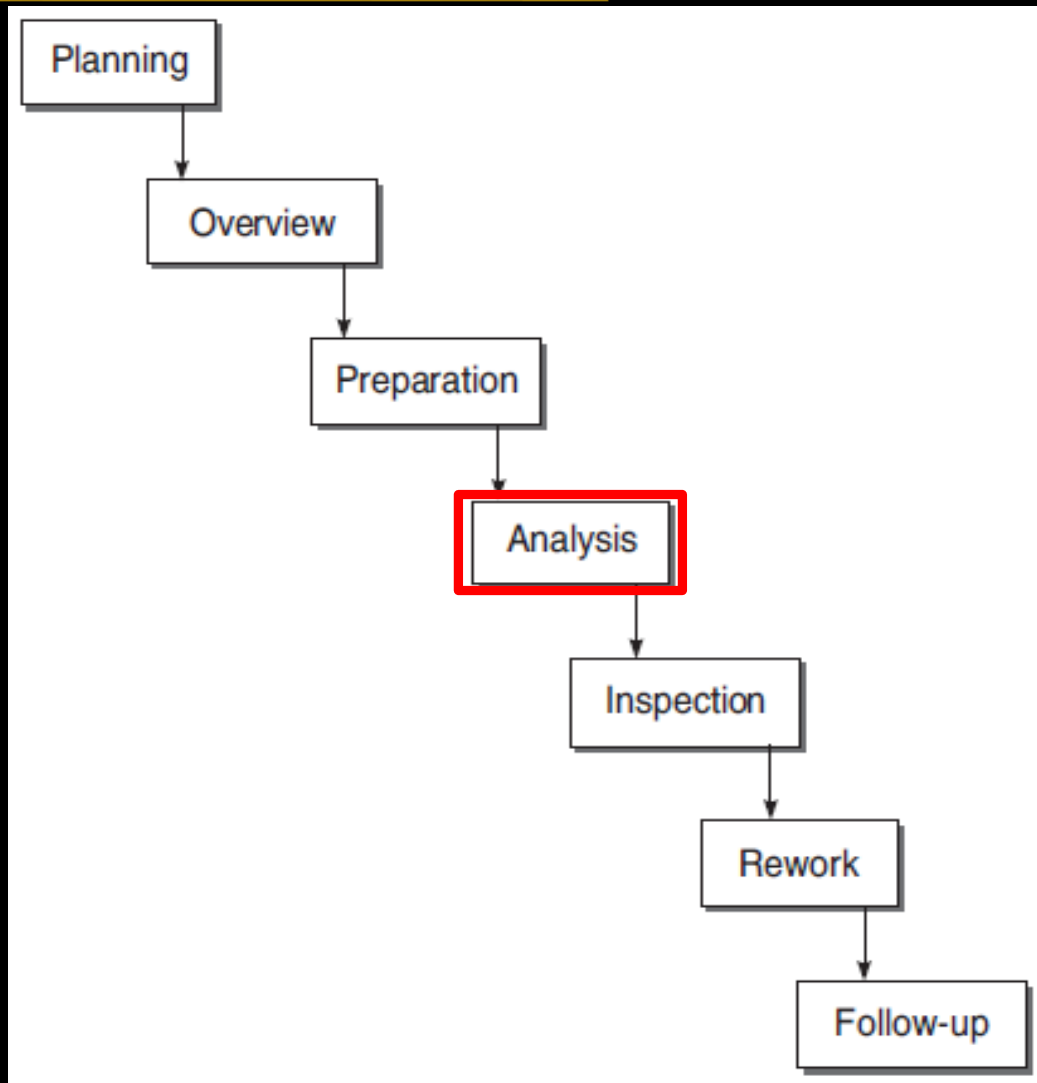
Follow-up, The leader **ensures** that the **edit phase** has been **executed properly**.

Exit, The inspection must **pass** the **exit criteria** as **fixed** for the completion of the inspection process.

Humphrey's Inspection Process

Preparation phase emphasizes the finding and logging of bugs, unlike Fagan inspections. It also includes an analysis phase wherein individual logs are analysed and combined into a single list.

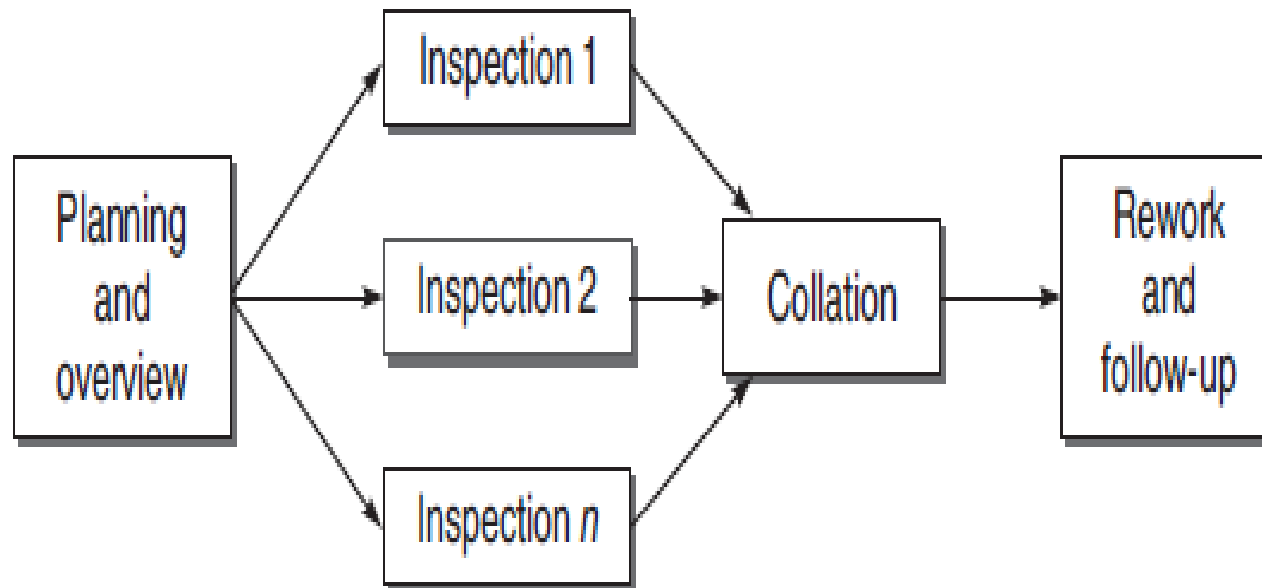
The steps of this process are shown below:



N-Fold inspections

Inspection process's effectiveness can be increased by replicating it by having multiple inspection teams.

Planning and overview: It includes the planning of **how many teams will participate** in the inspection process.



Inspection stages: There are many inspection processes adopted by many teams.

It is not necessary that every team will choose the same inspection process. The team is free to adopt any process.

Collation phase: The results from each inspection process are gathered, collated, and a master list of all detected defects is prepared.

Rework and follow-up: This step is same as the tradition Fagan inspection process.

Phased Inspection

Phased inspections are designed to verify the product in a particular domain by experts in that domain only.

Experts who have experience in that **particular domain** are called for inspection.

There are **two types of phases**,

(i) Single inspector

(ii) Multiple inspector

Single inspector: In this phase, a **rigorous checklist** is used by a single inspector to verify whether the **features specified** are there in the **item to be inspected**.

Multiple inspector:

There are **many inspectors** who are distributed the required **documents** for **verification** of an item.

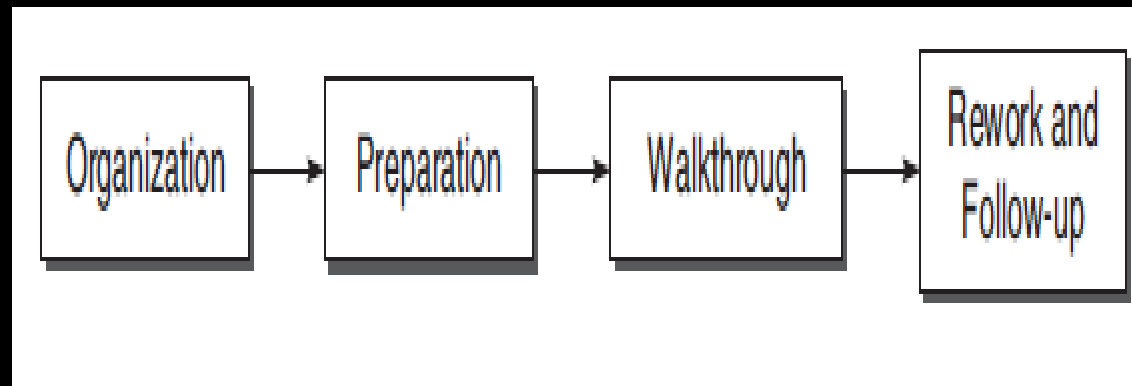
The **item** is then **inspected individually** by all the inspectors based on a **self-developed checklist** which is either **application or domain** specific.

After **individual checking** by the inspectors, a **reconciliation meeting** is organized where inspectors **compare their findings** about the item.

Structured Walkthrough

Described by Yourdon. Less formal and rigorous than formal inspections. Roles are coordinator, scribe, presenter, reviewers, maintenance oracle, standards bearer, user representative. Process steps are Organization, Preparation, Walkthrough, and Rework. Lacks data collection requirements of formal inspections.

A typical **structured walkthrough team** consists of the following members:



Coordinator: Organizes, moderates, and follows up the walkthrough activities.

Presenter/Developer: Introduces the **item to be inspected**. This member is optional.

Scribe/Recorder: Notes down the **defects found** and **suggestion proposed** by the members.

Reviewer/Tester: Finds the **defects** in the item.

Maintenance Oracle: Focuses on **long-term implications** and **future maintenance** of the project.

Standards Bearer: Assesses **adherence to standards**.

User Representative/Accreditation Agent: Reflects the **needs and concerns** of the user.

Object Oriented Testing

Testing an OO software is **more challenging**, Most of the **testing concepts lose their meaning** in OO technology,

e.g. a **unit** in OOT is **not a module** but a **class**, and a class **cannot be tested** with input-output behavior as in **unit testing** of a module.

Thus, both **testing strategies** and **techniques** change in case of OO software.

STRATEGY AND TACTICS OF TESTING OOS

Object-oriented software testing is generally done **bottom-up** at **four levels**:

(1) **Method-level testing**: It refers to the **internal testing** of an **individual method** in a **class**.

(2) **Class-level testing** : Class-level (or **intra-class**) testing refers to the testing of interactions among the **components** of an **individual class**.

(3) **Cluster-level testing**: Cluster-level (or **inter-class**) testing refers to the testing of **interactions among objects**.

(4) **System-level testing**: System-level testing is concerned with the **external inputs and outputs** visible to the **users** of a system.

Feature-based Testing of Classes

The **features of a class** can be categorized into **six main groups**:

(i) Create: These are also known as **constructors**. These features perform the initial **memory allocation** to the **object**, and initialize it to a **known state**.

(ii) Destroy: These are also known as **destructors**. These features perform the final **memory de-allocation** when the object is no longer required.

(iii) Modifiers: The features in this category **alter the current state** of the object.

Feature-based Testing of Classes

(iv) Predicates: The features in this category **test the current state** of the object for a **specific instance**. Usually, they return a **BOOLEAN** value.

(v) Selectors: The features in this category **examine and return** the **current state** of the object, or any part thereof.

(vi) Iterators: The features in this group are used to **allow all required sub-objects to be visited** in a **defined order**.