

OBJECT ORIENTED METHODOLOGIES:-

It is a new system approach which helps in the reuse of software components. It employs international Standard Unified Modelling Language (UML) from Object Management Group (OMG). Using this methodology a system can be developed on a component basis which enables the effective reuse of existing components, it facilitates the sharing of its other system components.

3 types: i) Object Modelling Technique (OMT)

ii) Object Process Methodology (OPM)

iii) Rational Unified Process (RUP)

OMT:- It is one of the 1st object oriented methodologies which uses 3 different models that are combined in a way that is analogous to the order structured methodologies.

Analysis:- The goal is to build models of world.

The requirements of the users, developers and managers provide the information needed to develop the initial problem statement.

OMT model:-

Object model: represents static structure of system.

Dynamic model: captures behaviour of system over time.

functional model: describes data transformation of system.

Design:-

Specifies all the details needed to describe how the system will be implemented and the system is designed.

OPM: It has one diagram which is used for modelling the structure functions & behaviour of system.

3 main process

eliciting:- determine higher level requirements, the scope of system & requirements and resources that are required.

developing:- involves detailed analysis, design & implementation of system.

deploying:- introduces the system to user & subsequent maintenance of system.

RUP:- consist of 4 phases 1. Inception.

2. Elaboration.

3. Construction.

4. Transition.

→ Each iteration consist of nine work areas of discipline.
→ A discipline depends upon the phase in which iteration is taking place.

OBJECTIVES:-

- encourage greater re-use.
- produce more detailed specification of system constraints.
- to have future problem with validation.

BENEFITS:-

- easier to produce & understand designs.
- It allows changes more easily.
- simplicity, reusability.
- Increased Quality, Maintenance.
- Scalable, Modularity, Modifiability.
- client/server architecture.

SOFTWARE QUALITY ASSURANCE:- is a process which works parallel to development of software. It focuses on improving the process of development of software so that problem can be prevented before they become a major issue. It has → Quality management approach.

- Formal technical reviews.
- Multi tasking strategies.
- Effective software technology.
- Measurement & Reporting mechanism.

ACTIVITIES:-

SQA Management Plan:- Make a plan for how you will carry it out with SQA throughout the project. Check level & SQA team skills.

Set the checkpoints:- SQA team should set checkpoints. Evaluate the performance of the project on basis of collected data on different checkpoints.

Multitasking Strategy:-

Do not depend on a single testing approach. When you have a lot of testing approaches available.

Measure Change Impact:-

The changes for major correction of an error ~~which~~ sometimes reintroduces more errors keep the measure of impact of change on project.

Manage Good Relations:-

In the working environment managing good relations with other teams involved in the project development is mandatory.

BENEFITS:-

- SQA produces high quality software
- High quality application saves time & cost
- SQA is beneficial for better reliability
- SQA is beneficial in the code & no maintenance for a long time
- High quality commercial software increases market share of company
- Improving the process of creating software
- Improves the quality of software.

Disadvantages of SQA:-

There are number of disadvantages of quality assurance some of them include adding

more resources employees more workers do help maintain quality & so much more.

Impact of object orientation on Testing:-

ERRORS -

There plausible (not worth testing for)

More plausible (worth testing for new)

New types of error may appear

Most current software testing techniques are congruent with functional software. A unit of software is either tested against its specifications or against some Code Coverage criterion to execute its identified path.

In object oriented system, testing encompasses three levels, unit testing, subsystem testing, system testing.

UNIT Testing:- In this, the individual cases are tested; i.e. whether the class attributes are implemented as per design and whether the methods and the interface are error free. Unit testing is the responsibility of application engineer.

SUBSYSTEM TESTING:- This involves testing a particular module or a subsystem and is the responsibility of the subsystem lead. It involves testing the associations within the system as well as the interaction of subsystem with the outside.

SYSTEM TESTING:- System testing involves testing the system as a whole and is the responsibility of the quality assurance team. The team often uses system tests as registered test when assembling new releases.

DEVELOP TEST CASES AND TEST PLANS:-

A test case is a document that lays out the following for a singular test scenario.

- test data
- scenarios
- description
 - procedures / inputs
 - testing environment
 - expected and actual results.

A test plan is a comprehensive document that lay out all major activities associated with a particular testing project. It includes.

- Scope of project
 - target market
 - goals & objectives.
 - assumption..
- testing environment.
- deliverables
 - testing cycle start/end dates
 - major risks & handling
 - testing tool data.

3 types:-

- i) Master test plan: has multiple levels of testing
- ii) phase test plan: address any one phase
- iii) Specific test plan: designed for non functional testing like security testing.

Steps to write a test plan Guidelines:

- | | |
|--------------------------------|-------------------------------|
| → analyse the product | → specifics |
| → design test strategy | → avoid redundancy |
| → define test objectives. | → avoid lengthy paragraphs |
| → define test criteria. | → use list & tables |
| → Resource planning | → delete unnecessary sections |
| → plan test environment | → update the plan. |
| → schedule and estimation. | |
| → determine test deliverables. | |

Importance:-

→ helps determine necessary effort to validate the quality of software application.

→ helps people understand the test details to outsiders like customers, developers, business managers.