Submission date: 10/01/2022

Mullapudi Renuka Devi
Batch-P, CSE
2019103037

OOAD - Assignment

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 the 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:
1. Object Modelling Techniques (OMT)
2. Object Process Methodology (OPM)
3. 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 older structured methodologies

Analysis:
- The goal is to build models of the world
- The requirements of the users, developers and managers provide the information needed to develop the initial problem statement

OMT models:
- Object Model: represents static structure of system
- Dynamic Model: Captures behaviour of system over time
- Functional Model: describes data transformations of system

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

2019103037

OPM: It has one diagram which is used for modelling the structure, function & behaviour of the system.
3 main process
- initiating: determines 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 the user & subsequent maintainance of system.

RUP: consists of 4 phases
1. Inception
2. Elaboration
3. Construction
4. Transition.

→ Each iteration consists 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 fewer problems with validation

BENEFITS:
→ easier to produce & understand designs
→ It allows changes more easily
→ Simplicity, Reusability
→ Increased Quality, Maintainance
→ Scalable, Modularity, Modifiability
→ Client/Server Architecture

2019103037

SOFTWARE QUALITY ASSURANCE: is a process which works parallel to development of software. It focusses on improving the process of development of software so that problems can be prevented before they become a major issue. It has
- Quality management approach
- Formal technical reviews
- Multi-testing 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 of SQA team skills

Set the Check Points: SQA team should set checkpoints. Evaluate the performance of the project on basis of collected data on different check points

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

Measure change Impact:
The changes for making correction of an error 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.

2019103037

BENEFITS :

→ SQA produces high quality software
→ High Quality application saves time & cost
→ SQA is beneficial for better reliability
→ SQA is beneficial in the condition of 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, employing more workers to help maintain quality & so much more

Impact of object Orientation on Testing

ERRORS :

- Less plausible (not worth testing for)
- More plausible (worth testing for now)
- New types of errors 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 paths. In object-oriented system, testing encompasses three levels, unit testing, subsystem testing, system testing

UNIT TESTING: In this, the individual cases are tested. It is seen whether the class attributes are implemented as per design and whether the methods and the interfaces 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 the 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 regression tests 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
→ descriptions
→ procedures / inputs
→ testing environment
→ expected and actual results

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

→ scope of project
→ target market
→ goals & objectives
→ assumptions
→ testing environment
→ deliverables
→ testing cycle start/end date
→ major risks & handling
→ testing end 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

→ analyse the product
→ design test strategy
→ define test objectives
→ define test criteria
→ Resource planning
→ plan test environment
→ schedule and estimations
→ Determine test deliverables

GUIDELINES

→ specificity
→ avoid redundancy
→ avoid lengthy paragraphs
→ use lists & tables
→ delete unnecessary sections
→ update the plan

Importance:
→ helps determine necessary effort to validate the quality of software application
→ helps people understand the test details related to the outside like customers, developers, business managers