Name: Md. Nasarul Hasan
Roll: 18
Home Task for Module-4: System Analysis and Design (ICT-103)]

1. What is SDLC? Briefly explain the phases of SDLC

Software Development Life Cycle (SDLC) is a process that defines the steps involved in the development a high-quality software. It covers the detailed plan for building, deploying and maintaining the software in a systematic and disciplined manner.

The explanation phases of SDLC is given below:

a) Requirement gathering and analysis

Planning for the quality assurance requirements and recognition of the risks involved is also done at this stage.

b) Design

The system and software design documents are prepared as per the requirement specification document.

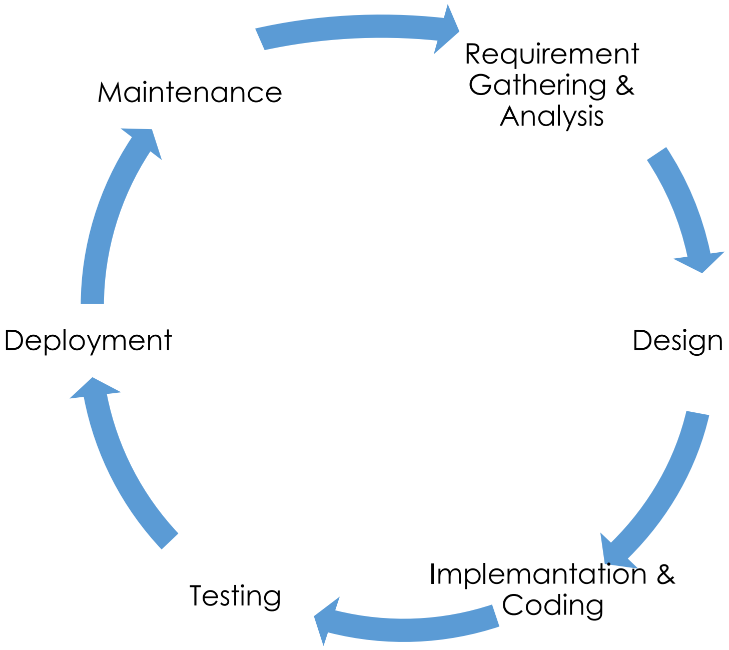


***Figure:** Software Development Life Cycle*

c) Implementation or coding

Developers start build the entire system by writing code using the programming language.

d) Testing

The testing team starts testing the functionality of the entire system to verify the entire application works according to the customer requirement.

e) Deployment

Deployment in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation.

f) Maintenance

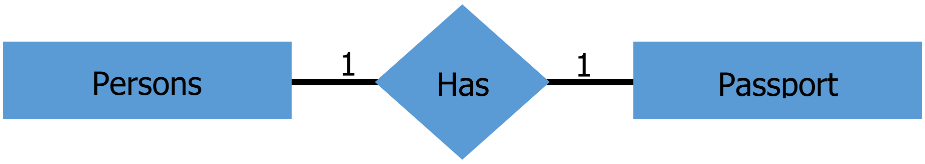
If any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

2. Show how the following relationships represented in ER model with suitable examples. (i) One-to-one, (ii) One-to-many & (iii) Many-to-many.

A relationship is represented by diamond shape in ER diagram, it shows the relationship among entities. Relationships in ER model shown below with a suitable example:

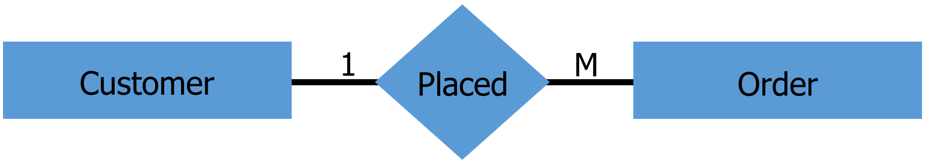
(i). One to One Relationship

When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship. For example, a person has only one passport and a passport is given to one person.



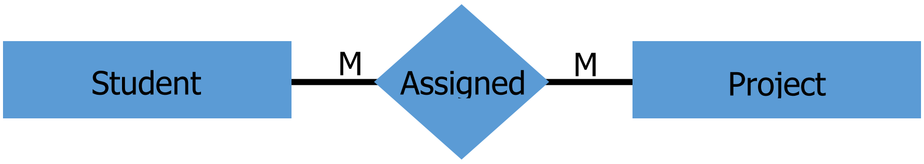
(ii). One to Many Relationship

When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship. For example – a customer can place many orders but an order cannot be placed by many customers.



(iii). Many to Many Relationship

When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship. For example, a student can be assigned to many projects and a project can be assigned to many students.



3. Summarize the advantages of using CASE tools in Systems Analysis and Design

The advantages of using CASE tools in systems analysis and design is given below:

a). Increased Speed

CASE tools provide automation and reduce time to complete tasks especially involving diagramming and associated specifications.

b). Increased Accuracy.

CASE tools can provide ongoing debugging and error checking which is very vital for early defect removal.

c). Reduced Lifetime Maintenance

Better design, analysis and automatic code generation, testing and debugging improves overall systems quality.

d). Better Documentation

Documentation are produce along the way because this tools have revisions for comments and notes on systems development and maintenance.

e). Programming in the hands of non programmers

By using the lower case tools, it could be possible to develop a software from the initial design and analysis phase.

f). Intangible Benefits

CASE tools can be used to allow for greater user participation, which can lead to better acceptance of the new system.

4. Briefly explain Unit Testing, Integration Testing and System Testing strategies

Unit testing

Unit testing is a type of software testing where individual units or components of a software are tested. The unit may be an individual function, method, procedure, module, or object.

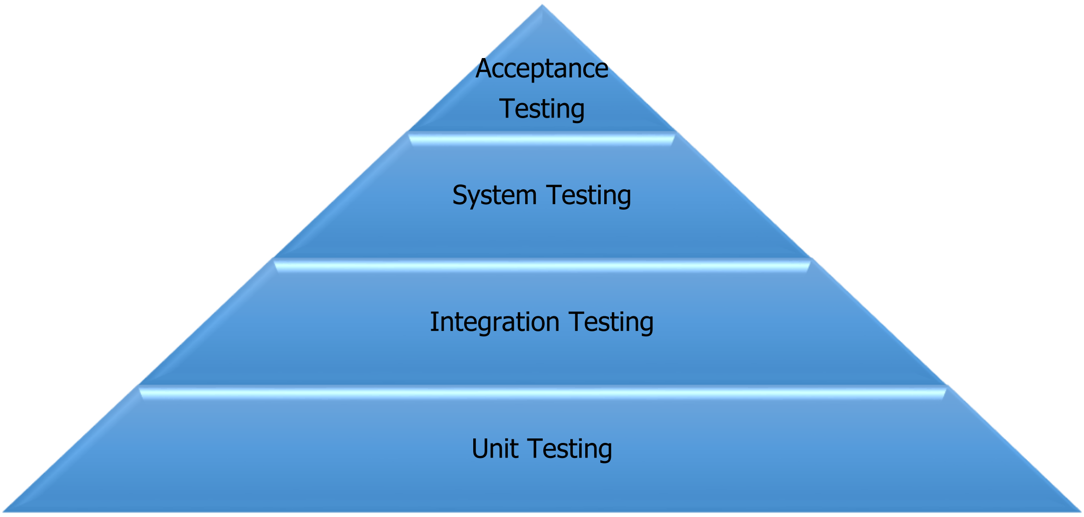


Figure: *Level of testing*

Integration testing

Integration testing is defined as a type of testing where software modules are integrated logically and tested as a group. Its focuses on checking data communication amongst these modules.

System testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications.

End