The Creators College of Science & Commerce Abbottabad

Lecturer Name: - Hassan Asghar

Subject Name: - Computer Science

Chapter No: - TWO [2]

Class: - 2st Year

LONG Questions

Question # 4: Define software development life cycle (SDLC). What are its objectives?

What is SDLC:

System Development Life Cycle (SDLC) is a step wise process of creating computer systems. It is also known as information system development or application development. It is a conceptual model which represents the necessary steps used for the development process of a software system. The SDLC is a problem-solving process which a series of steps helps to produce a new computer information system. The entire steps conducted in a sequence should provide the answers to a problem or opportunity. This step-wise procedure to build a system has a lot of importance. Some important points to remember are given following

- SDLC is important because it breaks down the entire life cycle of software development into phases thus making it easier for the development team members to easily evaluate each part of software development.
- SDLC makes it easier for programmers to work concurrently on each phase.
- It provides a rough time estimate that when the software will be available for use.
- It delivers quality software which meet or exceed customer expectations.
- It provides the basic framework for the developing of quality software.
- It ensures that the requirements for the development of the development of the software system are well defined and subsequently satisfied.

Objective of SDLC:

The objectives of the System Development Life Cycle (SDLC) are as follows:

- Delivery of quality software that meet the customer expectations.
- Delivery of inexpensive and cost-effective software which are easily maintainable.
- Maximize productivity in terms of the software systems delivered.
- One of the major objectives of SDLC is to establish an appropriate level of management authority to direct, co-ordinate, review, and approve the software development project.
- Proper Documentation of all the requirements needed for the development of the new software system.

Question # 5: What is a system? Where exactly the testing activities begin in SDLC?

System:

The term system is originated from the Greek term systema, which means to "place together." It can be defined as a set of interrelated components having a clearly defined boundary that work together to achieve a common set of objectives.

A system can be developed by applying a set of methods, procedures and routines in a proper sequence to carry outs some specific task. When all these functions are applied to build software then the system will be called as a software system.

Testing:

The execution of a program to find its errors is called testing. Here, the bugs are identified in the programmed module. The purpose of testing is to evaluate an attribute or capability of a program or system and determine that whether it meets its required results Testing/verification the software is actually operating the software under controlled conditions. It is the process of checking the items for consistency by evaluating the results against pre-specified requirements.

Question # 6: Why software development life cycle is important for the development of software?

Importance of software development life cycle:

- SDLC is important because it breaks down the entire life cycle of software development not phase thus making it easier for the development team members to easily evaluate each per of software development.
- SDLC makes it easier for programmers to work concurrently on each phase.
- It provides a rough time estimate that when the software will be available for use.
- It delivers quality software which meet or exceed customer expectations.
- It provides the basic framework for the developing of quality software.
- SDLC helps the project managers to establish a project management structure to be followed strictly during the system development.
- SDLC clearly define and assigns the roles and responsibilities of all the involved parties.
- Its ensures that requirements for the development o the software system are well defined and subsequently satisfied.

Question # 7: Who are stakeholders of SDLC? Describe their responsibilities?

Stakeholders of SDLC:

Those entities which are either within the organization or outside of the organization that sponsor a project, or have an interest or have the intention to get it after its successful completion, or may have positive or negative influence in the project completion are called stakeholders. Project stakeholders include the customers, the user group, the project manager, the development team and the testers. All those who have some interest in the project can be considered as stakeholders of that project. The individuals as well as the organizations that are actively involved in the project, or whose interests may be affected as a result of project execution or project completion are the part of stakeholders. It is the duty of the project execution or project completion are the part of stakeholders. It is the duty of the project management team to identify the stakeholders, determine their requirements, expectations are managing their influence in relation to the requirements to ensure a successful project.

Responsibilities of Stakeholders:

The basic role of the stakeholders is:

- For the development of a software system, resource such as time, money, equipment etc. are needed which should be provided to the project team by the stakeholders.

- Stakeholders educate the developers about their business.
- They spend more time to provide information and clarify requirements to the analysts are developers.
- The stakeholders should be specific and precise about the requirements
- Make timely decisions.
- Respect a developer's assessment of cost and feasibility.
- Set requirement priorities.
- Review and provide timely feedback.
- Promptly communicate changes to requirements.

Question # 8: What is meant by the term software requirement? Differentiate between functional and non- functional requirements?

Requirements Engineering (RE) is a set of activities used to identify and communicate the purpose of a software system, and the contexts in which it will be used. Requirement engineering consists of the following steps.

- Requirement Gathering
- Requirement Validation
- Requirement Management

Requirements are of two type given below:

Functional Requirements:

Functional requirements are those requirements of a software system which describe a function of a software or its components. It includes calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Examples of functional requirements are:

Interface requirement, Business requirements, Regulatory/Compliance Requirements and security requirements.

Non-Functional Requirements:

Non-functional requirements are those requirements which specify criteria for the judgment of the operations of a system. It describes that how well the system performs its duties. Nonfunctional requirements are often called qualities of a system. These requirements depend upon the nature of the software.

Different types of non-functional requirement are:

- Accessibility Requirements
- Accuracy Requirements
- Backup and Recovery Requirements
- Memory Capacity Requirements
- Compatibility Requirements
- Error-Handling Requirements
- Maintainability Requirements
- Performance Requirements
- Security Requirements

Question # 9: Design a flowchart for the following algorithm?

sum = 0, N = 5

x = 1

while x≤ N do

sum = sum + x

x = x + 1

end while

print sum

The flow chart of the above algorithm is given below: