INFO 5100 - Application Engineering & Dev

Homework to Chapter 15

Submitted By:

Abdul Haseeb Khan

NUID: 002844724

khan.abdulh@northeastern.edu

In the following pages, I will answer the questions in the Midterm Exam, providing detailed responses. Your feedback is greatly appreciated, and I am open to any additional guidance or expectations you may have for this assignment.

Non-Programming Assignment

1. What is SDLC and what are its steps?

Ans: **SDLC** stands for **Software Development Life Cycle**, which is a systematic process for planning, developing, testing, and maintaining information systems.

The following are the main stages of SDLC:

a. **Planning:** The scope, objectives and the requirements of the organization or software project are defined, analyzed, prioritize and arranged.

b. **Analysis:** The system requirements are studied, structured and analyzed.

c. **Design:** The functional and technical, detailed specifications of all system elements are designed.

d. **Implementation:** The code is developed, documented, and tested.

e. **Maintenance:** New versions or releases of software along with documentation, training and support

f. **Obsolescence:** Decision of making the software obsolete when replacement is required.

SDLC is needed for the following reasons:

Organized Approach: SDLC provides a structured and organized approach to software development, ensuring that each phase is planned and executed systematically.

Quality Assurance: It helps in identifying and fixing defects early in the development process through various testing stages, contributing to the overall quality of the software.

Risk Management: By following a step-by-step approach, SDLC allows for effective risk management, enabling teams to identify and mitigate potential risks at each stage.

Documentation: SDLC encourages the creation of documentation at each stage, helping in understanding the system, maintaining it, and transferring knowledge to others.

A Simple Example:

Using the Waterfall Model of the SDLC, we can determine the SDLC of developing a calculator.

Planning Phase:

Define the objective: Develop a basic calculator application with addition, subtraction, multiplication, and division functionalities.

Identify the scope: The calculator will have a simple user interface with buttons for each operation.

Create a project plan with timelines, resources, and budget estimation.

Analysis Phase:

Gather and analyze user requirements for the calculator application.

Create detailed specifications, including the layout, button designs, and logic for each operation.

Design Phase:

Define the architecture: Choose Java Swing for a desktop application.

Create a detailed design of the calculator's layout, including buttons for numbers and operations.

Produce technical specifications for developers to follow.

Implementation (Coding) Phase:

Write the code for the calculator application based on the design and technical specifications.

Implement the user interface using Java Swing.

Maintenance Phase:

Regularly update the calculator application to fix any reported issues.

Provide ongoing support and address user-reported problems.

Obsolescence Phase:

Evaluate the relevance and usefulness of the calculator application.

If necessary, plan for the retirement or replacement of the calculator application with a more modern solution.

7. List and describe different SDLC types.

Ans: There are several SDLC models, each with its own set of methodologies, advantages, and disadvantages. Here are some common SDLC types:

Waterfall Model:

Sequential and linear model where progress is seen as flowing steadily downwards through phases like Conception, Initiation, Analysis, Design, Construction, Testing, and Maintenance.

Advantages: Simple and easy to understand. Well-suited for small projects with clear requirements.

Iterative Model:

Description: Development is broken down into smaller cycles or iterations, with each iteration producing a deliverable product. Each iteration builds upon the previous one.

Advantages: Allows for feedback and adjustments during development. Useful when requirements are expected to change.

Incremental Model:

Description: Similar to iterative model, but focuses on building a complete and fully functional system by developing components or increments. Each increment adds new functionality.

Advantages: Allows for partial implementation and testing. Easier to accommodate changes and enhancements.

V-Model (Verification and Validation Model):

Description: Extends the waterfall model with corresponding testing phases for each development stage. The development and testing activities are performed in parallel.

Advantages: Emphasizes testing at every stage. Ensures early detection of defects.