Kareem Osama AlSayed

Task1
Report about SDLC

Software Development Life Cycle
SDLC

SDLC is a process used by software developers to analysis, design, develop, and test high-quality software. It aims to deliver software the meets customer expectations. It includes multiple phases:

A. System Requirement and Analysis:
   1. Objectives:
      - Understanding the business needs, features which software will provide and problems that the software will solve.

   2. Activities:
      - Product Owner gathers information from the client about the business requirements. He can support client with more features which leads to a utility between each other; so Product Owner should have sales experience.

   3. Result:
      - These requirements are then documented in a Software Requirement Specification SRS.

B. System Design and Implementation:

1. UI/UX:
   a. UX:
      1. Objectives:
         - Build a primary interface with which client can be comfortable.

      2. Activities:
         - Using different tools like “Figma and Adobe XD” and experience different interfaces to achieve his goal.

      3. Result
         - Having a structure design on which UI designer can work.

   b. UI:
      1. Objectives:
         - Complete the user interface design.

      2. Activities:
         - Build user interface with good design depending on the structure he got from UX designer to reach an application interface can be implemented.

      3. Result
         - UI design can be coded.

2. Front End:
      1. Objectives:
         - Build the application Views.
      2. Activities:
         - Front End developers build the pages depending on the UI design using different tools:
           - Languages:
             - HTML: Building structure of the pages.
             - CSS: Organizing structure and coloring it.
             - JAVASCRIPT: Making the pages dynamic and change between each other.

        - Framework:
            - There’s different frameworks like “Angular and React”, they are designed to reduce load times and improve the overall performance of a website, which can then lead to a better user experience.
    - After building pages Front End developers should integrate these pages with Back End API.

    3. Result:
        - User Interface of the application is ready.

3. Back End:
    1. Objectives:
        - Building a connection and controllers between views and database.

    2. Activities:
        - Back End developers are responsible for the connection between user interface(client side) and the database(server side), controlling operations from client on database through API and handling bugs and exceptions can happen.
        - He uses different tools divided to:
            - Programming language:
                - C#
                - JAVA
                - JAVASCRIPT
                - PHP
                - PYTHON

            - Frameworks:
                - .NET(C#)
                - SPRING(JAVA)
                - NODEJS(JAVASCRIPT)
                - LARAVEL(PHP)
                - DJANGO(PYTHON)

            - Object Relational Mapper(ORM)
                - ENTITY FRAMEWORK

- Functional Programming
    - LINQ(C#)

- Data Base Management System(DBMS)
    - SQL SERVER(C#)
    - ORACLE(JAVA)

3. Result:
    - Interconnection between database and client.

C. System Testing:

1. Objectives:
    - To ensure the software functions according to the requirements its free from bugs and errors.
2. Activities:
    a. TYPES:
    1. Unit Testing:
        - Testing individual components or modules to ensure they work correctly in isolation.
    2. Integration Testing:
        - Testing the interaction between modules to ensure they function together as expected.
    3. System Testing:
        - A complete test of the entire system to verify that all features work as intended.
    4. USER ACCEPTANCE TESTING (UAT):
        - Conducted with the end-users to ensure the software meets their needs and performs in real-world conditions.

    b. TOOLS:
    - Testers use both manual and automated testing methods, employing tools like “Selenium, JUnit, or TestNG” for automation.

3. Result:
    - Test plans, test cases, defect reports, and a tested version of the software ready for deployment

D. System Deployment and Maintenance:

1. Objects:
    - To release the software into the production environment where it can be used by the end-users and To ensure the software remains functional, up-to-date, and efficient over time.
2. Activities:
    - The software is installed on the users’ systems or on the cloud, depending on its nature and on different operating system using containers like DOCKER.

    - In the case of large-scale applications, deployment might be done in phases, such as beta testing or limited release, to gather feedback and make final adjustments.

    - Users may receive training, user manuals, or other documentation to help them understand and operate the system.
3. Result:
    - Users may receive training, user manuals, or other documentation to help them understand and operate the system.

    - Updated versions of the software, bug fixes, and performance enhancements.

SDLC Models:

Several SDLC Models exist, each has its different types of projects and organizational needs, one of most common models is:

- Agile:
    - The Agile model is an iterative approach that breaks down development into smaller cycles or "sprints." Teams work on small parts of the project in each sprint and gather feedback to necessary adjustments in the next iteration.
    - Advantages:
        - Flexible and adaptive; encourages frequent communication and collaboration; responds well to changing requirements.
    - Disadvantage:

- Requires high customer involvement; less predictable in terms of time and cost.