ID: 27381

ASSIGNMENT 1

Write a brief summary on Software Development Life Cycle. Explain each phase.

**Software Development Life Cycle (SDLC)**

The **Software Development Life Cycle (SDLC)** is a systematic process used by software engineers and project managers to design, develop, test, and maintain software systems. It provides a structured approach to software development, ensuring quality, efficiency, and predictability. The SDLC consists of several distinct **phases**, each with specific tasks and deliverables.

**Phases of SDLC**

1. **Requirement Gathering & Analysis**
   * **Objective**: Understand what the client or end user needs from the software.
   * **Activities**: Meetings with stakeholders, analyzing business needs, documenting requirements.
   * **Output**: Requirement Specification Document (SRS – Software Requirement Specification).

This is like figuring out what the client wants. Think of it as asking a bunch of questions to understand what kind of software they need and what problems it should solve. Once everything is clear, it's written down so everyone is on the same page.

1. **Design**
   * **Objective**: Plan how the system will work based on the requirements.
   * **Activities**: Architecture design, user interface design, database design, component design.
   * **Output**: Design Document Specification (DDS), prototypes, architecture diagrams.

Now that we know what’s needed, it’s time to plan how the software will look and work. This includes drawing out blueprints for the screens, how users will interact with it, and how the pieces will fit together behind the scenes.

1. **Implementation / Coding**
   * **Objective**: Convert the design into a functional software product.
   * **Activities**: Developers write code in the chosen programming language(s).
   * **Output**: Working software modules, versioned codebase.

This is where the actual building happens. Developers write code based on the design to bring the software to life. It’s like putting together all the parts of a machine to make it run.

1. **Testing**

* **Objective**: Ensure the software is error-free and meets requirements.
* **Activities**: Unit testing, integration testing, system testing, user acceptance testing.
* **Output**: Test cases, bug reports, verified and validated software.

After coding, the software is tested to find and fix any bugs or problems. It's like checking if all parts of the machine work correctly before handing it over.

1. **Deployment**

* **Objective**: Deliver the final product to users or the client environment.
* **Activities**: Installing the software, configuring servers, providing access.
* **Output**: Live or production-ready software.

Once it’s tested and ready, the software is delivered to the users or uploaded to a server where people can start using it. This is like launching the product into the real world.

1. **Maintenance**

* **Objective**: Update the software to fix bugs, improve performance, or meet new requirements.
* **Activities**: Patching, upgrades, monitoring, and customer support.
* **Output**: Updated software versions and continued user satisfaction.

Even after launch, the job isn’t done. There might be updates, small issues, or new features needed later. Maintenance is about keeping the software running smoothly over time.

