Assignment 1: SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment),

highlighting the importance of each phase and how they interconnect.

**Software Development Life Cycle (SDLC) Overview**

**1. Requirements:**

* **Importance:** Defines project scope, objectives, and functionalities.
* **Interconnect:** Provides the foundation for design and development phases by outlining project goals and user needs.

**2. Design:**

* **Importance:** Translates requirements into a detailed blueprint of the system architecture, interfaces, and functionalities.
* **Interconnect:** Guides the implementation phase by providing a roadmap for developers to follow.

**3. Implementation:**

* **Importance:** Involves coding, unit testing, and integration of system components.
* **Interconnect:** Builds upon the design phase to create the actual software product according to specifications.

**4. Testing:**

* **Importance:** Validates the software against requirements to ensure quality, functionality, and reliability.
* **Interconnect:** Feedback from testing informs developers of defects and areas for improvement, leading to iterative cycles of development and testing.

**5. Deployment:**

* **Importance:** Involves releasing the software to users or customers.
* **Interconnect:** Transition from development to production environment, marking the completion of the SDLC and the beginning of maintenance and support.

Assignment 2: Develop a case study analyzing the implementation of SDLC phases in a real-world engineering project. Evaluate how Requirement Gathering, Design, Implementation, Testing, Deployment, and Maintenance contribute to project outcomes.

**Case Study: SDLC Implementation in a Project Management Tool**

**Project Overview:** Developing a web-based project management tool for an engineering firm.

**SDLC Phases Implementation:**

**Requirement Gathering:**

* Interviews and surveys to define user needs and project objectives.
* Outcome: Detailed requirements document.

**Design:**

* Wireframes, mockups, and system architecture diagrams.
* Outcome: Finalized design documents.

**Implementation:**

* Agile development for iterative building.
* Outcome: Core functionalities developed.

**Testing:**

* Rigorous testing including unit, integration, and UAT.
* Outcome: Identified and fixed issues.

**Deployment:**

* Staging deployment and user training sessions.
* Outcome: Successful deployment to production.

**Maintenance:**

* Established support team for ongoing updates and fixes.
* Outcome: Continued usability and effectiveness.

**Evaluation:**

* Requirements set project direction.
* Design ensured usability and scalability.
* Implementation followed agile for flexibility.
* Testing maintained quality.
* Deployment was smooth.
* Maintenance ensured long-term relevance.

Assignment 3: Research and compare SDLC models suitable for engineering projects. Present findings on Waterfall, Agile, Spiral,and V-Model approaches, emphasizing their advantages, disadvantages, and applicability in different engineering contexts.

**Waterfall Model:**

* **Advantages:**
  + Clear project milestones and deliverables.
  + Well-defined requirements reduce ambiguity.
  + Easy to manage and understand.
* **Disadvantages:**
  + Limited flexibility for changes.
  + High risk of late-stage defects.
  + Not suitable for complex or evolving projects.
* **Applicability:** Small-scale projects with stable requirements and low risk.

**Agile Model:**

* **Advantages:**
  + Flexibility to accommodate changes throughout development.
  + Continuous feedback loops enhance product quality.
  + Suitable for dynamic and rapidly evolving projects.
* **Disadvantages:**
  + Requires active stakeholder involvement.
  + May lead to scope creep without proper management.
  + Less documentation can pose challenges for large teams or regulated industries.
* **Applicability:** Dynamic projects where requirements are likely to change.

**Spiral Model:**

* **Advantages:**
  + Iterative approach allows for risk management and early prototyping.
  + Flexibility to accommodate changes during development.
  + Well-suited for large-scale and complex projects.
* **Disadvantages:**
  + Requires significant resources and expertise.
  + Higher upfront costs due to prototyping and risk analysis.
  + Can be time-consuming and may lack clear project milestones.
* **Applicability:** High-risk projects with evolving requirements and a need for frequent prototypes.

**V-Model:**

* **Advantages:**
  + Emphasizes verification and validation throughout the development lifecycle.
  + Clear mapping between requirements and corresponding test cases.
  + Suitable for projects with strict regulatory compliance requirements.
* **Disadvantages:**
  + Linear approach may not accommodate changes well.
  + Requires comprehensive documentation and upfront planning.
  + Can be rigid and time-consuming for small-scale projects.
* **Applicability:** Projects with stringent quality assurance and compliance needs, such as in aerospace or medical industries.