Assignment-3

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.

1. **Waterfall SDLC Model**:

   The water model is one of the oldest and most straightforward approaches to software development.

   Phases:

   - **Requirements**: Gather requirements from stakeholders and analyze them to understand the project scope and objectives.
   - **Design**: Create a detailed design document outlining software architecture, user interface, and system components.
   - **Development**: Implement the software based on design specifications, including unit testing.
   - **Testing**: Test the software as a whole to ensure it meets requirements and is defect-free.
   - **Deployment**: Deploy the tested and approved software to the production environment.
   - **Maintenance**: Fix issues post-deployment and ensure ongoing compliance with requirements.

   Advantages:

   - **Simplicity**: Linear and sequential nature makes it easy to understand and implement.

   Disadvantages:

   - **Rigidity**: Changes are difficult once a phase is completed.
   - **Limited adaptability**: Not suitable for dynamic or evolving requirements.

   Applicability: Well-suited for stable projects with clear requirements.

2. **Agile SDLC Model**:

   Agile is not a specific methodology but rather a set of principles and values outlined in the Agile Manifesto. The Agile Manifesto prioritizes individuals and interactions, working solutions, customer collaboration, and responding to change over rigid processes and documentation

Phases:

- **Sprints**: Short development cycles with continuous feedback and adjustments.

Advantages:

- **Adaptability**: Easily accommodates changing requirements.
- **Collaboration**: Frequent communication among team members.

Disadvantages:

- **Complexity**: Requires active participation and coordination.
- **Documentation**: Minimal formal documentation.

Applicability: Ideal for dynamic projects where requirements evolve.

3. **Spiral SDLC Model**:

The Spiral model combines the idea of iterative development with the systematic aspects of the Waterfall model. It is based on the concept of a spiral, with each loop representing a phase in the software development process. The model is inherently risk-driven, meaning that risks are continuously assessed and addressed throughout the development life cycle.

Phases:

- **Planning**: Define objectives, constraints, and risks.
- **Engineering**: Develop and test the software.
- **Evaluation**: Review progress and assess risks.
- **Risk Analysis**: Identify and address potential risks.

Advantages:

- **Risk Management**: Explicit focus on risk assessment.
- **Flexibility**: Iterative approach allows for adjustments.

Disadvantages:

- **Complexity**: Requires skilled project management.
- **Time-Consuming**: Multiple iterations can extend the timeline.

Applicability: Suitable for large, complex projects with high risks.

4. **V-Model (Verification and Validation Model)**:

The V-Models, also known as the Verification and Validation models, is an extension of the traditional Waterfall models. It introduces a parallel testing phase for each corresponding development stage, forming a V-shaped diagram. Let's delve into the key principles that underpin the V-Models.

Phases:

- **Requirements**: Define requirements.
- **Design**: Create system design.
- **Coding**: Implement the design.
- **Testing**: Verify and validate at each level.

Advantages:

- **Thorough Testing**: Rigorous testing throughout the process.
- **Traceability**: Clear mapping between requirements and testing.

Disadvantages:

- **Rigidity**: Similar to Waterfall in terms of inflexibility.
- **Documentation Overhead**: Detailed documentation required.

Applicability: Well-suited for critical systems with strict quality requirements.