DAY 2:

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.

ANSWER:

Let's compare four common SDLC (Software Development Life Cycle) models: Waterfall, Agile, Spiral, and V-Model. Each of these models has distinct characteristics that make them suitable for different engineering contexts.

1. Waterfall Model

Overview:

- Sequential and Linear approach: Progresses through defined phases in a linear fashion.

- Phases: Requirements, Design, Implementation, Testing, Deployment, Maintenance (in that order).

- Diagram:

```
  Requirements   Design   Implementation   Testing   Deployment   Maintenance

       |           |        |          |        |           |

       V           V        V          V        V           V
```

Advantages:

- Clear structure and well-defined phases.

- Easy to manage due to its rigid structure.

- Suitable for projects with stable requirements and well-understood technologies.

Disadvantages:

- Limited flexibility for changes once a phase is completed.

- High risk of project failure if requirements are not properly gathered initially.

- Not ideal for projects where requirements are likely to evolve or where rapid iterations are needed.

Applicability: Suitable for projects with clear, stable requirements where changes are unlikely and where a sequential approach is appropriate (e.g., infrastructure projects, certain hardware development projects).

2. Agile Model

Overview:

- Iterative and Incremental approach: Develops software in small, incremental cycles (sprints).

- Key Principles: Collaboration, flexibility, rapid iterations, continuous improvement.

- Diagram:

```
  Requirements/Design  Implementation  Testing  Deployment

       |               |          |        |

       V               V          V        V
```

Advantages:

- Flexibility to accommodate changes throughout the development process.

- Customer satisfaction through early and continuous delivery of valuable software.

- Enhanced communication and collaboration among team members and stakeholders.

Disadvantages:

- Requires active customer involvement and frequent feedback, which can be challenging.

- Initial setup can be time-consuming due to the need for continuous integration and testing.

- May result in scope creep if not managed properly.

Applicability: Ideal for projects with evolving requirements, where rapid deployment of partial functionalities is beneficial, and where customer feedback plays a crucial role (e.g., software development, mobile app development).

3. Spiral Model

Overview:

- Iterative and Risk-Driven approach: Emphasizes risk analysis and mitigation throughout the development lifecycle.

- Phases: Planning, Risk Analysis, Engineering (development and testing), Evaluation (customer evaluation of the iteration).

- Diagram:

```
 Planning     Risk Analysis      Engineering     Evaluation

   |            |             |           |

   V            V             V           V
```

Advantages:

- Incorporates risk management throughout the project lifecycle.

- Allows for early identification and mitigation of risks.

- Flexibility to accommodate changes and refinements during each iteration.

Disadvantages:

- Can be complex and costly due to the need for extensive risk analysis.

- Requires a highly skilled team to effectively manage risks and iterations.

- Not suitable for small projects or projects with stable requirements.

Applicability: Suitable for large-scale projects where risks are high, and early risk identification and mitigation are critical (e.g., complex software projects, defense projects).

4. V-Model

Overview:

- Sequential and Verification and Validation (V&V) approach: Corresponds testing phases with development phases in a sequential manner.

- Phases: Requirements Specification, System Design, Architecture Design, Module Design, Unit Testing, Integration Testing, System Testing, Acceptance Testing.

Diagram:

```
 Requirements Specification  System Design  Architecture Design  Module Design

        |              |           |            |

        V              V           V            V

     Unit Testing     Integration Testing   System Testing    Acceptance Testing
```

```
|           |        |         |
V           V        V         V
```

Advantages:- Emphasizes rigorous testing and validation at each phase of development.

- Provides clear documentation and traceability between requirements and tests.

- Suitable for projects with defined and stable requirements.

Disadvantages:

- Can be inflexible and difficult to accommodate changes once development begins.

- Requires a structured and detailed upfront planning phase.

- May lead to increased project duration due to its sequential nature.

Applicability: Suitable for projects where requirements are well-understood and unlikely to change, and where thorough testing and validation are critical (e.g., medical device development, safety-critical systems).

Conclusion

Each SDLC model offers distinct advantages and is suitable for different engineering contexts based on project requirements, risk tolerance, and development approach. Choosing the right model depends on factors such as project size, complexity, expected changes, and criticality of requirements. Using the appropriate diagram for each model helps visualize their structure and phases, aiding in understanding their applicability and advantages in real-world engineering projects.