

# SDLC Phases

## Requirements

- Define project scope and goals, Identify user needs and expectations.
- Gather and document requirements, Critical for project success.

## Design

- Translate requirements into visual representations, Develop architecture and user interface.
- Create prototypes and mock-ups, Lays the foundation for implementation.

## Implementation

- Write code and develop software, Integrate components and systems.
- Conduct unit testing and debugging, Brings design to life.

## Testing

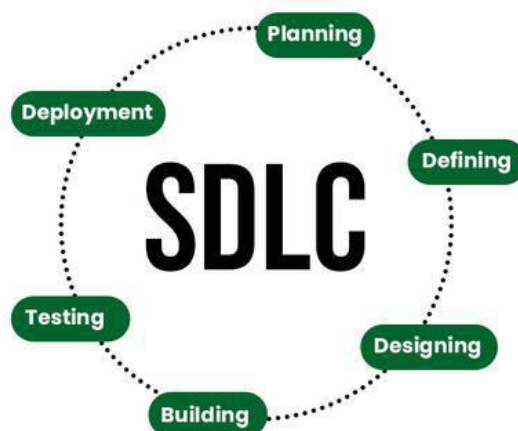
- Verify software meets requirements, Identify and fix defects.
- Ensure quality and reliability, Crucial for user satisfaction.

## Deployment

- Release software to production, Configure and install systems.
- Monitor and maintain performance, Delivers software to end-users.

## Interconnections

- Requirements inform Design, Design guides Implementation.
- Implementation is tested in Testing, Testing ensures smooth Deployment.



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.

# SDLC Phases and Implementation

## Requirement Gathering

- Stakeholder interviews and surveys to identify needs and expectations
- Analysis of existing traffic management systems and city infrastructure
- Definition of functional and non-functional requirements

**Outcome:** Clear understanding of project scope, stakeholder expectations, and system requirements

## Design

- Development of a detailed system architecture and design document
- Selection of technologies and tools for implementation
- Definition of system interfaces and data flows

**Outcome:** A well-defined system design, meeting stakeholder requirements and expectations

## Implementation

- Development of software and hardware components
- Integration of sensors, cameras, and other devices
- Configuration of network infrastructure and data storage
- Testing and debugging of individual components

**Outcome:** A functional system, meeting the defined requirements and design specifications

## Testing

- Unit testing, integration testing, and system testing
- Performance, security, and scalability testing
- Identification and fixing of defects and bugs
- Certification and acceptance testing

**Outcome:** A tested and validated system, ensuring quality and reliability

## Deployment

- Installation and configuration of the system in the production environment
- Data migration and initialization
- Training and support for end-users and maintenance teams

**Outcome:** A successfully deployed system, meeting stakeholder expectations and requirements

## Maintenance

- Ongoing monitoring and evaluation of system performance
- Identification and fixing of defects and bugs
- Updates and enhancements to the system
- Continuous improvement and optimization

**Outcome:** A maintained and improved system, ensuring long-term sustainability and effectiveness

### **Evaluation:**

The implementation of SDLC phases in this project contributed significantly to its success. The requirement gathering phase ensured a clear understanding of stakeholder needs, while the design phase resulted in a well-defined system architecture. The implementation phase produced a functional system, and the testing phase ensured quality and reliability. The deployment phase resulted in a successfully deployed system, and the maintenance phase ensured ongoing improvement and optimization.

# SDLC Models

## Waterfall Model

### - Advantages:

- Simple and easy to manage
- Phases are completed in a linear sequence
- Easy to measure progress

### - Disadvantages:

- No flexibility or iteration
- High risk if requirements are incomplete or inaccurate
- Testing is done late in the cycle

### - Applicability:

- Suitable for projects with well-defined requirements
- Projects with low complexity and risk

## Agile Model

### - Advantages:

- Flexible and adaptable
- Iterative approach with continuous improvement
- Early detection of defects and errors

### - Disadvantages:

- Requires high team collaboration and communication
- Difficult to measure progress
- Can be chaotic if not managed properly

### - Applicability:

- Suitable for projects with changing or unclear requirements
- Projects with high complexity and risk
- Projects requiring rapid delivery and iteration

## Spiral Model

### - Advantages:

- Combines benefits of Waterfall and Agile
- Iterative approach with incremental releases
- Risk assessment and mitigation

### - Disadvantages:

- Complex to manage and control
- Requires high expertise and resources
- Can be time-consuming and costly

**- Applicability:**

- Suitable for projects with high complexity and risk
- Projects requiring rapid delivery and iteration
- Projects with multiple stakeholders and requirements

## V-Model

**- Advantages:**

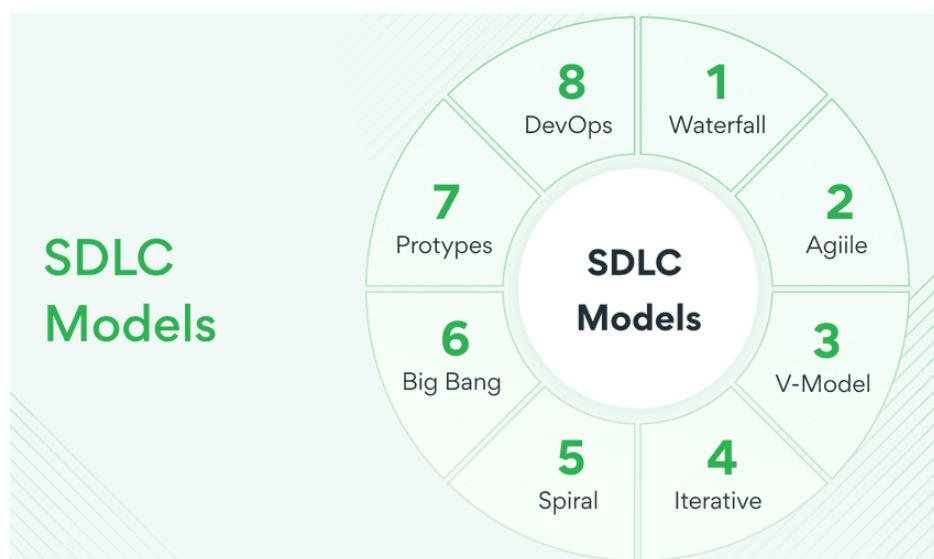
- Emphasizes testing and validation
- Verification and validation at each stage
- High quality and reliability

**- Disadvantages:**

- Rigid and inflexible
- Testing is done late in the cycle
- High risk if requirements are incomplete or inaccurate

**- Applicability:**

- Suitable for projects with high safety and reliability requirements
- Projects with complex systems and integration



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.