System Development Methodologies and Their Life Cycles:

Structured Analysis and Design (SAD):

- Life Cycle:

1. Feasibility Study: Assess project viability and constraints.

2. System Planning and Selection: Define objectives, scope, and methodologies.

3. Systems Analysis: Gather and document user requirements.

4. System Design: Create detailed specifications and system architecture.

5. Implementation: Translate design into code and develop the system.

6. Testing: Ensure the system functions correctly through various testing phases.

7. Deployment and Installation: Roll out the system into the production environment.

8. Maintenance and Support: Provide ongoing updates and support.

9. Documentation: Maintain comprehensive project documentation.

10. Post-Implementation Review: Assess project success and gather lessons learned.

- Elements:

- Essential Model: Defines system functionality without implementation details.

- Environmental Model: Establishes system boundaries and interactions.

- Components include the Statement of Purpose, Context Diagram, and Event List.

- Behavioral Model: Describes internal system behavior and data entities.

- Components include Data Dictionary, Data Flow Diagram, Entity Relationship Diagram, Process Specification, and State Transition Diagram.

- Implementation Model: Maps functional requirements to hardware and software.

- Defines human-computer interfaces and non-functional requirements.

- Utilizes Structure Charts.

- Advantages: Visual, mature, process-oriented, flexible, and simple.

- Disadvantages: Limited user-analyst interaction, challenges in determining when to stop decomposition.

Object-Oriented Analysis and Design (OOAD):

- Life Cycle:

1. Analysis Phase: Develop a model of the real-world application with functional behavior.

2. Design Phase: Refine the analysis model, focusing on overall architecture and implementation details.

3. Implementation Phase: Translate the design into code using programming languages or database management systems.

- Elements:

- Objects: Represent real-world entities, encapsulating data and behavior.

- Classes: Define structure and behavior of objects.

- Inheritance: Allows classes to inherit attributes and behaviors from others.

- Polymorphism: Enables objects of different classes to be treated as a common superclass.

- Encapsulation: Hides internal details and exposes necessary interfaces.

- Advantages: Thoroughly represents complex relationships, consistent notation, focuses on essential features.

- Disadvantages: Complexity in early stages, requires a learning curve for developers.

In summary, both Structured Analysis and Design (SAD) and Object-Oriented Analysis and Design (OOAD) follow specific life cycles and encompass various elements. SAD's life cycle includes feasibility assessment, planning, analysis, design, implementation, testing, deployment, maintenance, documentation, and review. OOAD's life cycle involves analysis, design, and implementation, with a focus on objects, classes, inheritance, polymorphism, and encapsulation. Each methodology has its own strengths and weaknesses, making them suitable for different project scenarios.