OBJECT-ORIENTED SYSTEM ANALYSIS AND DESIGN

BY: YUMESH BAN

DATE : 25 JANUARY

INTRODUCTION TO OOSAD

Definition and Purpose:

- Object-Oriented System Analysis and Design (OOSAD) is a methodology that focuses on using objects as the primary units of analysis and design.

Importance in Modern Software Development:

- It facilitates more natural modeling of real-world entities
- Promotes reusability and maintainability in software development.

TRADITIONAL VS. OBJECT-ORIENTED APPROACH

- **Traditional:**
  - Process-centric or data-centric problem decomposition
- **Object-oriented:**
  - Balances process and data by focusing on objects containing both.

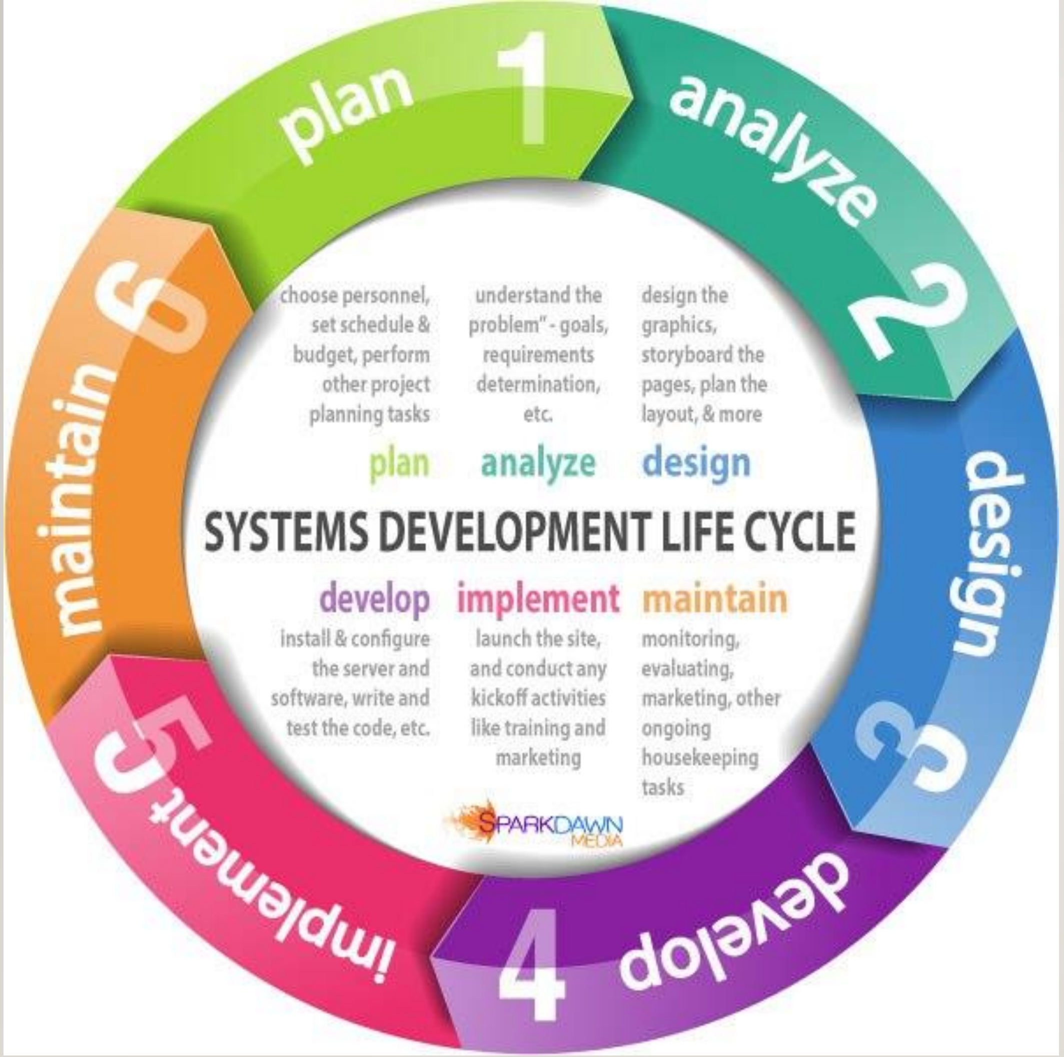
EMERGENCE OF OBJECT-ORIENTED METHODOLOGIES

- Difficulty in modeling real-world systems with traditional approaches.
- Emergence of RAD-based object-oriented methodologies
- Focus on objects that contain data and processes

OOSAD NATURE REPRESENTATION:

Object Oriented Design is the process for demonstrating logical and physical as well as static and dynamic models of the system under the design so it requires SDLC.

SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)



SYSTEMS DEVELOPMENT LIFE CYCLE

plan: choose personnel, set schedule & budget, perform other project planning tasks

analyze: understand the problem" - goals, requirements determination, etc.

design: design the graphics, storyboard the pages, plan the layout, & more

develop: install & configure the server and software, write and test the code, etc.

implement: launch the site, and conduct any kickoff activities like training and marketing

maintain: monitoring, evaluating, marketing, other ongoing housekeeping tasks

- Phases of SDLC:
  - ❖Planning
  - ❖Analysis
  - ❖Design
  - ❖Implementation
  - ❖Maintenance

CORE PRINCIPLES OF OOSAD

- Use case driven
- Architecture centric
- Iterative and incremental development

USE CASE DRIVEN APPROACH

- Definition and importance of use cases
- Use cases as primary modeling tool
- Simplicity in focusing on one activity at a time

ARCHITECTURE CENTRIC APPROACH

- Importance of system architecture in specification, construction, documentation
- Three architectural views: Functional, Static, Dynamic

FUNCTIONAL VIEW

- External behavior of the system from user perspective
- Relation to process-modeling in structured analysis

STATIC VIEW

- Structure of the system: attributes, methods, classes, relationships, messages
- Relation to data-modeling in structured analysis

DYNAMIC VIEW

- Internal behavior of the system: messages between objects, state changes within objects
- Combination of process and data-modeling approaches

ITERATIVE AND INCREMENTAL DEVELOPMENT

- Continuous testing throughout the project lifecycle
- System evolves to meet user needs with each iteration

BENEFIT OF EACH APPROACH

- **Use Case Driven Benefits**:
  - ✓Improved communication between users and analysts
  - ✓Better understanding of user needs
- **Architecture Centric Benefits**:
  - ✓Comprehensive system understanding from multiple views
  - ✓More complete depiction of information system
- **Iterative and Incremental Development Benefits**:
  - ✓Continuous testing and refinement
  - ✓Higher quality systems that meet user needs

KEY CONCEPTS AND THEIR BENEFITS

- **Classes , Objects & Methods**:

  Realistic modeling, better communication

- **Encapsulation**:

  Loosely coupled units, fewer ripple effects.

- **Inheritance**:

  Less redundancy, faster creation of new classes

- **Polymorphism**:

  Simpler event programming, ease in replacing objects

BENEFITS OF OOSAD

- Realistic modeling of business processes
- Improved communication between users and developers
- Reusable objects and system components

CHALLENGES IN OOSAD

- **Complexity**:

  Managing and designing complex systems can be challenging.

- **Learning Curve**:

  Requires understanding of object-oriented concepts and UML.

- **Integration**:

  Integrating with existing systems and legacy code can be difficult.

BEST PRACTICES

- **Tips for Effective OOSAD**:
  - **Clear Requirements Gathering**: Ensure requirements are well-defined and understood.
  - **Consistent Use of UML**: Use UML diagrams consistently for documentation.
  - **Iterative Development**: Use iterative and incremental development approaches to manage complexity.

CONCLUSION

- We learned about the OOSAD principles and benefits.
- We also reviewed the importance of OOSAD in modern information systems development
- We emphasized the benefits, such as reusability, scalability, and maintainability.