**Difference between Structured and Object-Oriented Analysis**

In the realm of software engineering, the analysis phase plays a pivotal role in understanding the requirements of a system and devising effective strategies for its design and implementation. Two prominent methodologies used for this purpose are Structured Analysis and Object-Oriented Analysis. While both approaches aim to dissect and comprehend the intricacies of a system, they differ significantly in their principles, techniques, and conceptual frameworks.

## What is Structured Analysis?

Structured analysis is a methodology that emphasizes breaking down a system into smaller, more manageable components. It relies on graphical tools, primarily Data Flow Diagrams (DFDs), to illustrate how data flows through the system. Furthermore, it employs a Data Dictionary to define data elements and their relationships, providing a structured view of the system’s functionality.

## What is Object-Oriented Analysis?

Object-Oriented Analysis (OOA) revolves around modeling a system as a collection of interacting objects. Objects encapsulate both data and behavior, mirroring real-world entities and their interactions. OOA emphasizes the identification of classes, objects, attributes, and methods, facilitating a more modular, reusable, and intuitive design approach.

Below are the differences between structures and object oriented analysis:

| **Structured Analysis** | **Object-Oriented Analysis** |
| --- | --- |
| The main focus is on the process and procedures of the system. | The main focus is on data structure and real-world objects that are important. |
| It uses System Development Life Cycle (SDLC) methodology for different purposes like planning, analyzing, designing, implementing, and supporting an information system. | It uses Incremental or Iterative methodology to refine and extend our design. |
| It is suitable for well-defined projects with stable user requirements. | It is suitable for large projects with changing user requirements. |
| Risk while using this analysis technique is high and reusability is also low. | Risk while using this analysis technique is low and reusability is also high. |
| Structuring requirements include DFDs (Data Flow Diagram), Structured Analysis, ER (Entity Relationship) diagram, CFD (Control Flow Diagram), Data Dictionary, Decision table/tree, and the State transition diagram. | Requirement engineering includes the Use case model (find Use cases, Flow of events, Activity Diagram), the Object model (find Classes and class relations, Object interaction, Object to ER mapping), Statechart Diagram, and deployment diagram. |
| This technique is old and is not preferred usually. | This technique is new and is mostly preferred. |

**Conclusion:**

In conclusion, while both Structured Analysis and Object-Oriented Analysis serve the purpose of analyzing and designing software systems, they differ significantly in their approach and application. Structured Analysis provides a systematic breakdown of the system’s functionality, whereas Object-Oriented Analysis offers a more intuitive and flexible modeling approach. The choice between these methodologies depends on the nature of the project, its complexity, and the preferences of the development team.