Systems Analysis:

The process of analyzing a system or its components to understand its interconnections, functions, and goals is called systems analysis.

It is used to identify and understand system requirements and to create or improve systems.

The goal of systems analysis is to identify problems and inefficiencies in existing systems and to develop and implement solutions that improve the functionality and effectiveness of the overall system.

System analysis often involves several steps:

• Define the problem or opportunity.Clarify what the system analysis will focus on.

• Information Gathering: Gather data about the current system's inputs, outputs, operations, and environment.

• System Analysis: Examine collected data to find sources of problems, inefficiencies, or areas for development.

• Generate and evaluate viable solutions to identified problems or areas of improvement.

• Make changes: Implement the chosen solution and track its effectiveness over time to ensure it achieves the desired results.

Requirements and Functional Requirements: Functional requirements and system analysis requirements are two important steps in the system development process.

System analysis requirements are a broad set of statements that specify what the system should do and how problems should be handled.

To generate these needs, data about the current system and its stakeholders are collected and systematically analyzed.

Functional requirements, on the other hand, provide a detailed description of how the system will perform a particular activity or task.

These requirements define the required system functionality as well as system inputs, processing, and outputs.

Both types of needs are interdependent.

System analysis requirements specify the context and high-level goals of the system.

Functional requirements, on the other hand, describe the exact actions and tasks that the system must perform to achieve those goals.

System analysis requirements create functional requirements, and the system development process uses the functional requirements to design and build the system.

Both types of requirements are necessary to ensure that the completed system meets the needs and expectations of stakeholders.

Non-Functional Requirements: Software requirements that describe quality characteristics or limitations of a system are called non-functional requirements.

In contrast to functional requirements, non-functional requirements determine how well the system performs the task at hand.

Examples of non-functional requirements include: • Performance specifications: List the capabilities of the system in terms of response time, throughput, and scalability.

• Define the required level of system security, including data protection, authentication, and access control.

• Describe the system's user interface, navigation, and usability requirements.

• In the Compatibility Requirements section, specify the platforms, software, or hardware that your system must be compatible with.

• Maintainability Requirements: Describes how the system can be easily updated, maintained, and improved over time.

• Reliability requirements: indicates the level of system reliability required, including availability, fault tolerance, and recovery.

• Scalability Requirements: Describes the system's ability to handle growing groups of users or workloads.