

Introduction to System Analysis and Design

Introduction to System Analysis and Design

Key concepts, phases, and best practices for building effective information systems.

1. What Is System Analysis & Design?

Aspect	Definition
System Analysis	The process of examining a business problem or opportunity, gathering requirements, and defining what the new system must do.
System Design	Translating the analysis results into a blueprint (architecture, data models, interfaces, processes) that guides construction of the system.
Goal	Deliver a solution that meets user needs, fits the organization’s environment, and can be built, maintained, and evolved efficiently.

2. Why It Matters

- **Aligns technology with business goals** – prevents costly mismatches.
 - **Reduces risk** – early identification of constraints, feasibility, and stakeholder expectations.
 - **Improves quality** – structured documentation, review cycles, and testing plans.
 - **Facilitates communication** – shared artifacts (models, specifications) keep analysts, developers, and users on the same page.
-

3. Core Concepts & Terminology

Term	Meaning
Stakeholder	Anyone with an interest in the system (users, managers, regulators, IT staff).
Requirement	A condition or capability the system must satisfy (functional vs. non-functional).
Feasibility Study	Evaluation of technical, economic, operational, legal, and schedule viability.
Modeling	Visual representation of system aspects (processes, data, behavior).

Introduction to System Analysis and Design

Term	Meaning
Prototyping	Building an early, limited-function version to explore requirements and gather feedback.
Life Cycle Model	The sequence of phases a system goes through (e.g., Waterfall, Agile, Spiral).
Change Management	Controlling modifications to the system after deployment.

4. The System Development Life Cycle (SDLC)

4.1 Traditional (Waterfall) Model

1. **Planning** – Define scope, objectives, resources, schedule.
2. **Analysis** – Gather detailed requirements; produce *System Requirements Specification (SRS)*.
3. **Design** – Create *Logical* and *Physical* designs (data models, architecture, UI mockups).
4. **Implementation** – Code, integrate, and unit-test components.
5. **Testing** – System, integration, acceptance testing; verify against requirements.
6. **Deployment** – Install, train users, go-live.
7. **Maintenance** – Corrective, adaptive, perfective, preventive activities.

4.2 Agile / Iterative Models

- **Iterative cycles** (sprints) deliver small, usable increments.
- Emphasize **continuous stakeholder collaboration**, **adaptive planning**, and **frequent testing**.
- Common frameworks: Scrum, Kanban, Extreme Programming (XP).

4.3 Hybrid Approaches

- Combine **structured analysis** (for complex domains) with **agile delivery** (for flexibility).
 - Example: *Agile-Waterfall* – detailed upfront analysis, then iterative development.
-

5. System Analysis Phase

Introduction to System Analysis and Design

Activity

Typical Output

Stakeholder Identification	Stakeholder register, RACI matrix
Requirement Elicitation	Interview notes, questionnaires, workshops, observation logs
Requirement Documentation	Functional Requirements (FRs), Non-Functional Requirements (NFRs), Use Cases, User Stories
Feasibility Assessment	Feasibility report (technical, economic, operational, legal, schedule)