## The software development lifecycle

The 7 stages of the software development life cycle are Planning, Requirements, Design, Develop, Test, Deploy and Maintenance. In this document we are going to discuss the role of software design in the IT Systems Development Life Cycle and the importance of it.

## The Design Stage

The Software design stage plays a critical role in the Software Development Life Cycle (SDLC) by converting requirements and business goals into step-by-step instructions that guide the development process and reduces risks while laying the foundation for software that meets the needs and demands of all stakeholders.

## Role of Software Design in the SDLC

The design phase begins after the requirements stage, ensuring that the goal of the software is well understood, that all stakeholders have agreed with the full list of requirements (functional, non-functional, user and business) as detailed in the Software Requirements Specification (SRS) document.

Similarly, the design phase ends with a fully detailed design for the development stage. Essentially this means translating the list of requirements into detailed instructions i.e. UI/UX wireframes, Software Design Documents (SDDs), architecture/data diagrams, ER diagrams, technology lists etc. Ensuring that the software developers can produce a software solution in line with the original requirements.

## Importance of Software Design

Proper design prevents technical debt (bugs or poorly designed solutions that cannot be fixed without unexpected or undesirable software development work), uncovers potential bugs early, and avoids costly rework by clarifying the solution before coding even begins.

Design documents serve as roadmaps and/or blueprints, enabling coordination among team members and stakeholders, improving transparency and communication and ensuring the delivery of the most effective solution.

## Inputs and Deliverables

**Inputs**: Software requirements specification (SRS), task lists, stakeholder feedback, business goals, technical constraints, existing assets.

**Deliverables**:

* Software Design Document (SDD)
* UI/UX wireframes, mock-ups
* Architecture/data diagrams
* Technology list

**Role**: Bridges planning/requirements and implementation by turning requirements into actionable blueprints.

**Design Docs**:

* High-level: System overview for managers/architects
* Low-level: Detailed specs for developers
* Architecture diagrams: Visual blueprints for clarity and early issue detection