**System Analysis and Design**

System analysis and design is a process that involves understanding the requirements of a system and creating a plan for its development. It is a critical step in the software development life cycle, as it helps to ensure that the system meets the needs of its users and is developed efficiently and effectively.

**Example**: Online Shopping System

Consider an online shopping system, where customers can browse products, add items to their cart, and place orders. The system consists of various components such as a web application, a database to store product information and customer data, and a payment gateway to process transactions.

1. **Identify Components**: In system analysis, we first identify the main components of the system. For the online shopping system, these components might include:

* User interface (web application)
* Database server
* Payment gateway

1. **Understand Interactions**: Next, we analyse how these components interact with each other. For example:

* The web application retrieves product information from the database and displays it to users.
* When a customer places an order, the web application sends the order details to the database to update the inventory.
* The payment gateway processes payment transactions and updates the order status in the database.

1. **Study Functionality**: We examine the functionality of each component to understand how they contribute to the overall system. For instance:

* The web application allows users to search for products, view product details, and add items to their cart.
* The database stores product information, user accounts, and order history.
* The payment gateway securely processes payment transactions using credit cards or other payment methods.

1. **Identify Improvements**: System analysis involves identifying areas for improvement or optimization. For the online shopping system, potential improvements might include:

* Enhancing the user interface to improve navigation and usability.
* Optimizing database queries to improve performance and reduce response times.
* Implementing additional security measures to protect user data and prevent fraud.

1. **Propose Solutions**: Based on the findings of the analysis, we propose solutions to address identified issues or meet system requirements. For example:

* Upgrade the web application framework to a more modern and feature-rich platform.
* Implement caching mechanisms to reduce database load and improve response times.
* Integrate additional authentication and authorization mechanisms to enhance security.

1. **Implementation and Testing**: Finally, the proposed solutions are implemented, and the system is tested to ensure that it functions as expected and meets the defined requirements.

**SDLC Architecture**

The Systems Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying, and maintaining software. The SDLC is a process for planning, creating, testing, and deploying an information system. It applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.

The SDLC consists of seven stages:

1. **Requirement Analysis**:

In this initial phase, the project's requirements are gathered, analyzed, and documented. This involves understanding the needs of stakeholders, identifying system functionalities, and defining project scope.

1. **Planning**:

Once requirements are understood, the project is planned in detail. This phase involves creating project timelines, defining milestones, allocating resources, and estimating costs. Project planning sets the foundation for subsequent development phases.

1. **Design**:

In this phase, the system architecture, software design, and user interface are planned and documented. Design decisions made during this phase establish the structure of the software and ensure it meets the specified requirements.

1. **Implementation (Coding):**

The design is translated into code during the implementation phase. Developers write and test code according to the specifications outlined in the design phase. This step may involve multiple iterations to refine the software and address any issues that arise.

1. **Testing:**

The software is rigorously tested to identify and fix defects, errors, and bugs. Testing may include various techniques such as unit testing, integration testing, system testing, and acceptance testing. The goal is to ensure that the software functions correctly and meets the defined requirements.

1. **Deployment**:

Once the software has been thoroughly tested and approved, it is deployed to the production environment. This involves installing the software on target systems, configuring settings, and preparing for user access. Deployment may also involve data migration and user training.

1. **Maintenance and Support**:

After deployment, the software enters the maintenance phase. During this phase, ongoing support, bug fixes, and updates are provided to ensure the software remains functional and meets evolving user needs. Maintenance activities may include troubleshooting, performance optimization, and feature enhancements.

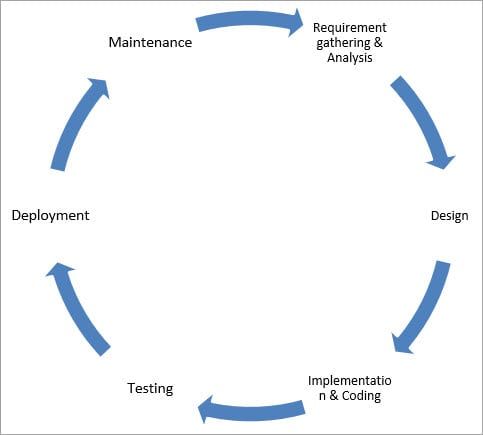


Fig: Phase of SDLC (Software Development Life Cycle)