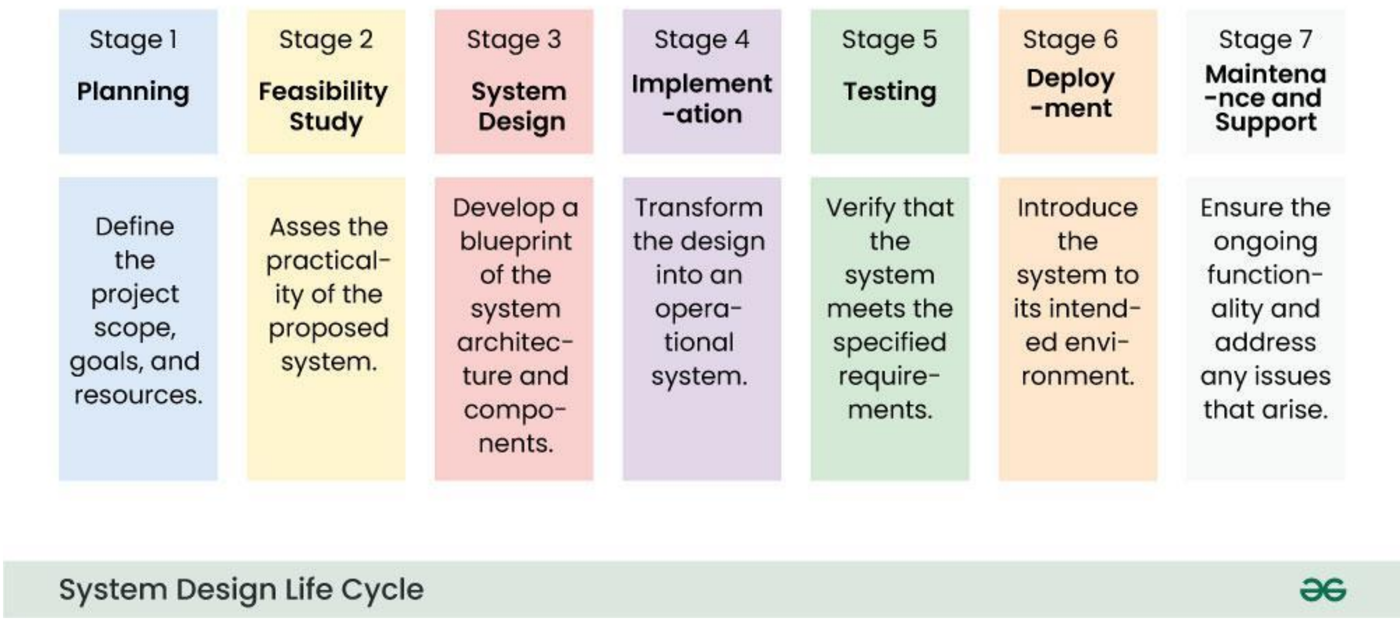
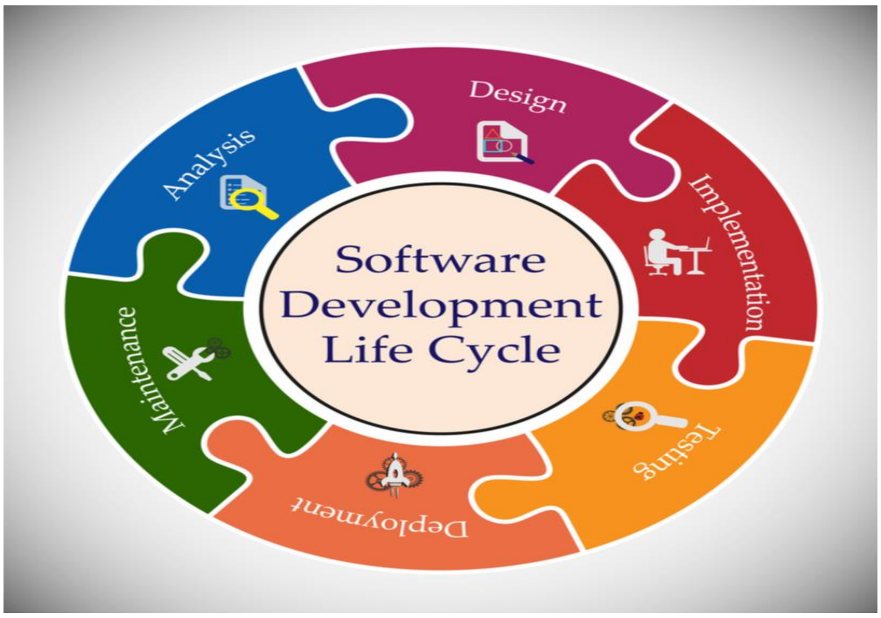
Assignment 1: SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.



Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6	Stage 7
Planning	**Feasibility Study**	**System Design**	**Implement -ation**	**Testing**	**Deploy -ment**	**Maintena -nce and Support**
Define the project scope, goals, and resources.	Asses the practical-ity of the proposed system.	Develop a blueprint of the system architec-ture and compo-nents.	Transform the design into an opera-tional system.	Verify that the system meets the specified require-ments.	Introduce the system to its intend-ed envi-ronment.	Ensure the ongoing function-ality and address any issues that arise.

System Design Life Cycle



Stage 1. Planning

- **Objective:** Define the project scope, goals, and resources.
- **Example:** Imagine a company initiating a project to develop a new customer relationship management (CRM) system. The planning phase would involve outlining the functionalities, budge constraints, and identifying the team responsible.

Stage 2. Feasibility Study

- **Objective:** Asses the practicality of the proposed system.
- **Example:** Before committing to the CRM project, a feasibility study would analyze factors like technical, operational, and economic viability. This involves evaluating whether the benefits outweigh the costs.

Stage 3. System Design

- **Objective:** Develop a blueprint of the system architecture and components.
- **Example:** For the CRM system, this involves creating a detailed design that outlines the database structure, user interfaces, and system functionalities. It serves as a guide for the developers during the coding phase.

Stage 4. Implementation

- **Objective:** Transform the design into an operational system.
- **Example:** Developers write the code for the CRM system based on the design specifications. This phase involves rigorous testing to identify and rectify any bugs or errors.

Stage 5. Testing

- **Objective:** Verify that the system meets the specified requirements.

- **Example:** The CRM system undergoes various testing procedures, such as unit testing, integration testing, and user acceptance testing, to ensure its functionality, performance, and security.

Stage 6. Deployment

- **Objective:** Introduce the system to its intended environment.
- **Example:** The CRM system is deployed for use by the organization’s employees. This may involves training sessions to familiarize users with the new system.

Stage 7. Maintenance and Support

- **Objective:** Ensure the ongoing functionality and address any issues that arise.
- **Example:** Regular updates, bug fixes, and user support for the CRM system to adapt to changing business requirements and address any emerging issues.

Differences between the System Development Life Cycle and the System Design Life Cycle

Aspect	System Development Life Cycle	System Design Life Cycle
Definition	A comprehensive framework covering the entire system development process.	A subset of the SDLC that specifically deals with designing the system.
Scope	Encompasses the entire life cycle of a system, form initiation to retirement.	Focuses primarily on the design aspects of the system.
Phases	Typically includes Planning, Analysis, Design, Implementation, Testing, Deployment, and Maintenance.	Usually includes Feasibility Study, System Analysis, System Design, Implementation, Testing, Deployment, and Maintenance.
Focus	Broad focus on the overall development process, addressing planning,	Specific focuses on the design phase, detailing how the system will be built and operate.

Aspect	System Development Life Cycle	System Design Life Cycle
	implementation, testing, and maintenance.	
Purpose	Guides the development team through the entire process, from concept to post-deployment support.	Provides a blueprint for constructing the system based on specified design requirements.

Challenges in System Design Life Cycle

- **Unclear Requirements:** Sometime, the initial requirements for a system might be unclear or ambiguous, leading to difficulties in designing the system accurately.
- **Changing Requirements:** Requirements may change during the design process, posing a challenge to maintain consistency and ensuring that the system still meets the user's needs.
- **Technological Changes:** Rapid advancements in technology can make it challenging to choose the most suitable and up-to-date technologies for system design.
- **Integration Issues:** Ensuring seamless integration of various system components can be complex, especially when dealing with different technologies and platforms.
- **Budget Constraints:** Designing a system within budgetary constraints can be challenging, as incorporating certain features or technologies might be cost-productive.
- **Security Concerns:** Designing a system that is secure from potential threats and vulnerabilities is an ongoing challenge, as new security risk continually emerge.
- **Scalability and Performance:** Designing a system to handle scalability and ensuring optimal performance, especially under heavy loads, can be challenging.
-

Models Used for System Design Life Cycle

- **Waterfall Model:** A linear and sequential model where each phase must be completed before moving on to the next. It's a straightforward approach but can be inflexible in the face of changing requirements.

- **Iterative Model:** Involves repeating cycles, with each iteration refining and improving the system based on feedback. It’s adaptable to changing requirements.
- **Prototyping Model:** Involves building a prototype (a preliminary version) of the system to gather feedback refine the design before building the final product.
- **Spiral Model:** Incorporates elements of both iterative and prototyping models. It involves cycles of planning, designing, constructing, and evaluating.

Conclusion

In conclusion, the System Design Life Cycle (SDLC) plays a pivotal role in shaping the development of robust and efficient systems. By focusing on the design aspects, it provides a blueprint for constructing systems that meet user requirements and adhere to industry standards.
Throughout the exploration of Software Design Life Cycle, we’ve highlighted key aspects, including the differences between the System Design Life Cycle and the broader System Development Life Cycle, the stages of the System Design Life Cycle, challenges faced during the process, commonly used models, best practices, and practical use cases.