SYSTEM DEVELOPMENT LIFE CYCLE

For years now, the Software Development Life Cycle, or SDLC for short, has cemented itself as the de-facto process to help build information systems, systems engineering, and software engineering from the ground up by encompassing key phases that can be grouped in planning, implementation, and maintenance of the system solution. The SDLC has grown to be critical thanks to its standardized phases that manage a balancing act between costs, quality, and time to meet modern business demands, urgency, complexity, and to top it off, with tight budgets.

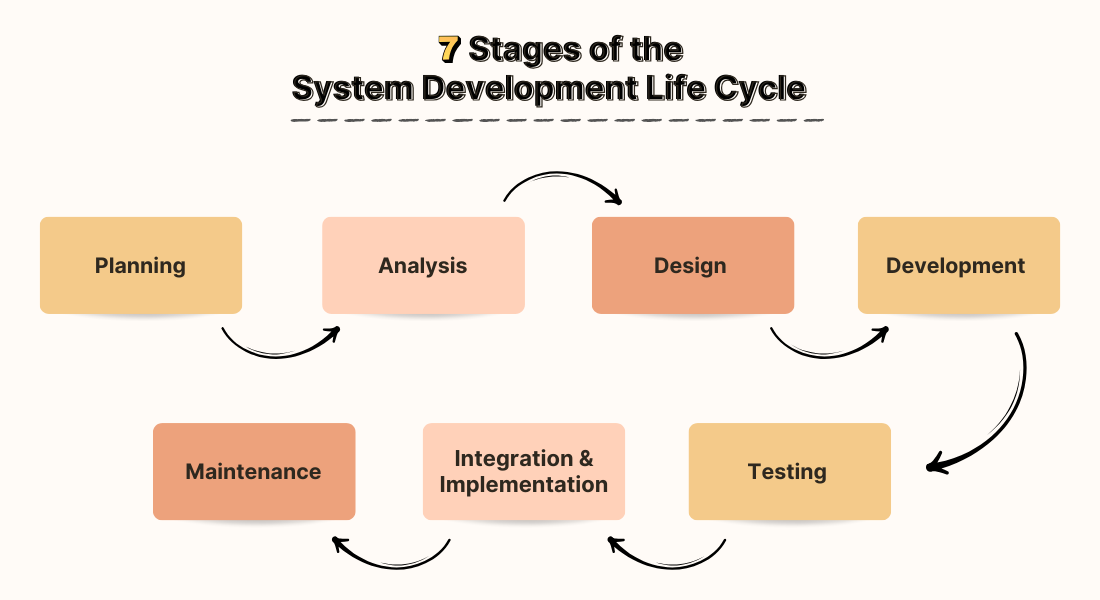


## WHAT IS SDLC?

SDLC is a closed loop in which each stage affects the actions in subsequent ones and provides clear information for future stages. To answer specific questions and ensure consistency in your development process, usually, all six stages try to effectively and consistently influence each other. But before we run and take off by explaining each of the SDLC phases, let’s first define what a system is. A system is a combination of hardware, software, and human resources that perform the assigned tasks of collecting, processing, and displaying information. Some may wonder why it’s so important to develop and operate information systems in a cyclical fashion and to that, we can find the answer as it’s traced back to the constant changes in conditions in which the information system is located that influence the outcome of the system. Whether it’s upgrading to new hardware, updating software development tools, increasing user requirements, or scaling the amount of information in the business environment and domain areas, a system’s life cycle always consists of planning, implementing, analysis, design, development, testing and maintaining. Now, we will cover the main phases involved in the System Development Life Cycle to review what each entails.

# THE SDLC PHASES:

The SDLC phases are designed in a way that progressively develops or alters a system across its life cycle. If followed through from beginning to end, the SDLC will help deploy a fully-operational, high-quality system that meets and/or exceeds client requirements, all within the specified time and budget constraints.



**1. PLANNING:**

Setting a strong foundation and defining a clear understanding of a project is crucial to the success of any information system. For this reason, the SDLCs first phase is planning where stakeholders and all parties involved in the project participate to clearly define requirements and the nature of what the information system will need to solve. The planning phase helps delineate all subsequent tasks so they can be planned and budgeted for accordingly. To achieve a comprehensive planning cycle, members of the project need to have a deep understanding of what tasks the future information system needs to solve. With that foundation as context, the quality and time spent on the planning phase have a direct correlation to the success of the project. In this phase, the team defines the key components of the project at a high level, they define the environment in which the information system will operate along with the necessary technical, budget, and human resources required to complete the project.

**2. ANALYSIS:**

Once a thorough plan is set in place, next comes the analysis phase. This crucial phase is where project members dive deep and define the technical requirements of the system so they can be properly addressed. The analysis phase in SDLC allows you to receive feedback and support from relevant internal and external stakeholders. At the same time, you will need to think broadly about who your potential users will be. At this stage, you will include your clients, designers, management team, programmers, testers, and other technical team members. In general, this stage is all about answering the question: “What problems need to be solved?” Also, during the analysis phase, the team defines the inputs and outputs of the data flow in and out of the system by undertaking a thorough system analysis of the business processes that need to be covered and solved by the future system. This phase is closely tied to documenting all the project specifications and the team usually takes sufficient time to properly document each detail for future reference.

**3. DESIGN:**

Progressing down the SDLC, the next phase that typically follows analysis is the design phase. In this phase, all the documentation that the team created in the analysis phase is used to develop the actual technical documentation of the project. In the design phase, project members define the structure of project components as well as key elements of the system by defining the interfaces that will exchange data within the workflow. It’s very common for the project teams to use UML diagrams in the design phase to design the system’s architecture.

**4. DEVELOPMENT:**

The development phase is used to write the actual code of the system’s software, develop and deploy the system’s hardware, configure cloud systems, implement interaction protocols, and prepare the primary test data. During the development phase, project teams carefully implement the system’s architecture in program code by creating methods and algorithms for information processing, preparing the project outputs, and building a monitoring system for the system.

**5. TESTING:**

Right after the development phase, comes testing and deployment. What does this phase entail, really? Well, for any system to work as intended, it needs to be thoroughly tested and tested again until the results match the expected outcome. This phase is crucial as it directly impacts the quality of the outputs.

This phase involves three key activities:

5(a): Test Plan Creation : Before testing the product, a detailed test strategy is created. This strategy outlines the testing technique, identifying which aspects of the product will be tested and how. A thoroughly developed test strategy is essential for a methodical and effective testing process.

5(b): Test Execution and Bug Fixing : Once the test plan has been established, the real testing of the product occurs. Analyzers document any flaws they discover, which are then sent to the engineers to be fixed. After the bugs have been fixed, the product is tested again to confirm that the fixes function and do not introduce new issues. This process of testing, repairing, and retesting continues until the product meets the quality standards outlined in the test plan.

5(c): User Acceptance Testing (UAT) : User Acceptance Testing (UAT) is an important stage in the product testing process in which genuine end users or clients test the product. The goal of UAT is to ensure that the product works well in real-world scenarios and satisfies the client’s specific needs and requirements.

**6. DEPLOYMENT:**

Once the testing is done, and the product is ready for deployment, it is released for customers to use. The size of the project determines the complexity of the deployment. The users are then provided with the training or documentation that will help them to operate the software. Again, a small round of testing is performed on production to ensure environmental issues or any impact of the new release.

**7. MAINTENANCE:**

Maintenance is an ongoing phase where the system is monitored, maintained, and updated as needed. This includes bug fixes, performance enhancements, security patches, and responding to user feedback. The actual problem starts when the customer actually starts using the developed system and those needs to be solved from time to time. According to the changing user end environment or technology, the software is updated timely. Proper maintenance ensures the system remains efficient, secure, and adaptable to future business needs.