What are the Software Development Life Cycle (SDLC) phases?

There are various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as "Software Development Process Models" (e.g. **Waterfall model**, **incremental model**, **V-model**, **iterative model**, **RAD model**, **Agile model**, **Spiral model**, **Prototype model** etc.). Each process model follows a particular life cycle in order to ensure success in process of software development.

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design which is called development phase. After coding and development the testing verifies the deliverable of the implementation phase against requirements. The testing team follows **Software Testing Life Cycle (STLC)** which is similar to the development cycle followed by the development team.

There are following seven phases in every Software development life cycle model:

1. Requirement gathering and analysis
2. Feasibility Study
3. Design
4. Implementation or coding
5. Testing
6. Deployment
7. Maintenance

1) **Requirement gathering and analysis:** Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements like; Who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.
   Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model. The testing team follows the Software Testing Life Cycle and starts the **Test Planning** phase after the requirements analysis is completed.

2) **Feasibility Study:** In case the system proposal is acceptable to the management, the next phase is to examine the feasibility of the system. The feasibility study is basically the test of the proposed system in the light of its workability, meeting user's requirements, effective use of resources and of course, the cost effectiveness. These are categorized as technical, operational, economic, schedule and social feasibility. The main goal of feasibility study is not to solve the problem but to achieve the scope. In the process of feasibility study, the cost and benefits are estimated with greater accuracy to find the Return on Investment (ROI). This also defines the resources needed to complete the detailed investigation. The result is a feasibility report submitted to the management. This may be accepted or accepted with modifications or rejected. In short, following decision are taken in different feasibility study:

   Economic feasibility - The likely benefits outweigh the cost of solving the problem which is generally demonstrated by a cost/ benefit analysis.

   Operational feasibility - Whether the problem can be solved in the user's environment with existing and proposed system workings?

   Organizational feasibility – Whether the proposed system is consistent with the organization's strategic objectives?

   Technical feasibility - Whether the problem be solved using existing technology and resources available?

   Social feasibility – Whether the problem be solved without causing any social issues? Whether the system will be acceptable to the society?

3) **System Design**: Based on the user requirements and the detailed analysis of a new system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the development of a system. The logical system design arrived at as a result of system analysis and is converted into physical system design. In the design phase the SDLC process continues to move from the what questions of the analysis phase to the how . The logical design produced during the analysis is turned into a physical design - a detailed description of what is needed to solve original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail. In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided. Data structure, control process, equipment source, workload and limitation of the system, Interface, documentation, training, procedures of using the system, taking backups and staffing requirement are decided at this stage.

There are several tools and techniques used for describing the system design of the system. These tools and techniques are: Flowchart, Data flow diagram (DFD), Data dictionary, Structured English, Decision table and Decision tree

4) **Coding:** The system design needs to be implemented to make it a workable system. This demands the coding of design into computer language, i.e., programming language. This is also called the programming phase in which the programmer converts the program specifications into computer instructions, which we refer to as programs. It is an important stage where the defined procedures are transformed into control specifications by the help of a computer language. The programs coordinate the data movements and control the entire process in a system. A well written code reduces the testing and maintenance effort. It is generally felt that the programs must be modular in nature. This helps in fast development, maintenance and future changes, if required. Programming tools like compilers, interpreters and language like c, c++, and java etc., are used for coding .with respect to the type of application. The right programming language should be chosen.

5) **Testing** :Before actually implementing the new system into operations, a test run of the system is done removing all the bugs, if any. It is an important phase of a successful system. After codifying the whole programs of the system, a test plan should be developed and run on a given set of test data. The output of the test run should match the expected results. Sometimes, system testing is considered as a part of implementation process**.**

Using the test data following test run are carried out:

- Program test
- System test

Program test : When the programs have been coded and compiled and brought to working conditions, they must be individually tested with the prepared test data. All verification and validation be checked and any undesirable happening must be noted and debugged (error corrected).

System Test : After carrying out the program test for each of the programs of the system and errors removed, then system test is done. At this stage the test is done on actual data. The complete system is executed on the actual data. At each stage of the execution, the results or output of the system is analyzed. During the result

analysis, it may be found that the outputs are not matching the expected output of the system. In such case, the errors in the particular programs are identified and are fixed and further tested for the expected output. All independent modules be brought together and all the interfaces to be tested between multiple modules, the whole set of software is tested to establish that all modules work together correctly as an application or system or package.

When it is ensured that the system is running error-free, the users are called with their own actual data so that the system could be shown running as per their requirements.

6) **Implementation:** After having the user acceptance of the new system developed, the implementation phase begins. Implementation is the stage of a project during which theory is turned into practice. The major steps involved in this phase are:

- Acquisition and Installation of Hardware and Software
- Conversion
- User Training
- Documentation

The hardware and the relevant software required for running the system must be made fully operational before implementation. The conversion is also one of the most critical and expensive activities in the system development life cycle. The data from the old system needs to be converted to operate in the new format of the new system. The database needs to be setup with security and recovery procedures fully defined.

During this phase, all the programs of the system are loaded onto the user's computer. After loading the system, training of the user starts. Main topics of such type of training are:

- How to execute the package?
- How to enter the data?
- How to process the data (processing details)?
- How to take out the reports?

After the users are trained about the computerized system, working has to shift from manual to computerized working. The process is called **Changeover**. The following strategies are followed for changeover of the system.

1. **Direct Changeover:** This is the complete replacement of the old system by the new system. It is a risky approach and requires comprehensive system testing and training.
2. **Parallel run** : In parallel run both the systems, i.e., computerized and manual, are executed simultaneously for certain defined period. The same data is processed by both the systems. This strategy is less risky but more expensive because of the following facts:

- Manual results can be compared with the results of the computerized system.
- The operational work is doubled.
- Failure of the computerised system at the early stage does not affect the working of the organization, because the manual system continues to work, as it used to do.

(iii) Pilot run: In this type of run, the new system is run with the data from one or more of the previous periods for the whole or part of the system. The results are compared with the old system results. It is less expensive and risky than parallel run approach. This strategy builds the confidence and the errors are traced easily without affecting the operations.

The documentation of the system is also one of the most important activity in the system development life cycle. This ensures the continuity of the system. Generally following two types of documentations are prepared for any system.

- User or Operator Documentation
- System Documentation

User Documentation: The user documentation is a complete description of the system from the user's point of view detailing how to use or operate the system. It also includes the major error messages likely to be encountered by the user.

System Documentation: The system documentation contains the details of system design, programs, their coding, system flow, data dictionary, process description, etc. This helps to understand the system and permit changes to be made in the existing system to satisfy new user needs.

7) **Maintenance :** Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environments. It must meet the scope of any future enhancement, future functionality and any other added functional features to cope up with the latest future needs. It has been seen that there are always some errors found in the systems that must be noted and corrected. It also means the review of the system from time to time. The review of the system is done for:

   •knowing the full capabilities of the system

   •knowing the required changes or the additional requirements

   •studying the performance.

Systems Development Life Cycle (SDLC) puts emphasis on decision making processes that affect system cost and usefulness. These decisions must be based on full consideration of business processes, functional requirements, and economic and technical feasibility. The primary objectives of any SDLC is to deliver quality system which meets or exceed customer expectations and within cost estimates, work effectively and efficiently within the current and planned infrastructure, and is an inexpensive to maintain. SDLC establishes a logical order of events for conducting system development that is controlled, measured, documented, and ultimately improved.