

# Elaborated Phases of Software Development Life Cycle

Radhika D Amlani  
Research Scholar  
Saurashtra University  
Rajkot Gujrat

Dr. Deepak Kumar  
District Informatics Officer  
(Scientific Officer/Engineer-SB)  
NIC, Porbandar

## ABSTRACT

Computer science is a young science. Computer programmers were trained by engineers. That training dictated how they approached software development for an entire generation. But now after decades of building software to be expensive, unwanted, and unreliable programmers have come to realize software is different. Building software is more like creating a work of art; it requires creativity in design and total craftsmanship to complete. In this paper, researcher tried to explain all the phases of Software Development Life Cycle in detail through which all the programmers have to go through to develop any software. The task of programmers would be easy.

## Keywords

*SDLC, Analysis, Designing.*

## 1. INTRODUCTION

Software Development Life Cycle Model is used as a process of creating and altering current existing system. SDLC used in information system, systems engineering, and software engineering. SDLC can be thought of as a concept that used by many software development methodologies, which are currently available in market or software industry. SDLC provide a framework to create, plan and control any information system to be developed.

Developers worldwide will agree that building software takes more than just writing complex codes and implementing them in an environment. Developers usually start out their career in programming by developing programs or software according to their own plan and hope that someone will appreciate it. But once the developer is associated with a business or another software company, the creativity is limited to business and consumer needs. The pressure in creating accurate and efficient software is even bigger in the entrepreneurial stage.

To ensure developers have come up with the right software for the specific need, programmers have created steps on how a program could specifically created. This will ensure everything is built according to plan and tested extensively before it could be implemented for public or formal use in business.

Under these circumstances the term "Systems Development Life Cycle" was born. The need to create accurate and efficient software has led to the formalization on certain stages and phases on how a program should be built.

Simply put, SDLC or Systems Development Life Cycle is a series of steps observed by developers on building specific software. Developers follow certain steps to ensure they have the right software for the right demand.

The history of the term "Systems Development Life Cycle" is not very clear but it naturally came into being since the 1960s

when developers started to create programs specific to a certain need. Slowly, the term has been observed by different software development companies. From a simple format of planning, building, testing and implementing, software companies have developed their own version of developing specific products for their clients. Each version of software development is called "Model".

SDLC describes a process, which is used by engineers and developers to create and deploy all features or characteristic of information system. This features or characteristic describes requirements, validation, training and emphasizing ownership of the system. Whenever SDLC is used, the goal is to create a system that meets the primary objective of the owner within defined timeline and cost constraints. It also includes post installation stages like, deployment and maintenance, which has features like ease of use, installation of the software, for minimizing error.

There are so many SDLC based, software engineering models available in market now-a-days. Depending upon the suitability, the software engineering model can be used to put forward any software project. Each of the methodologies or models has different level of risk and benefits to cope with the project requirements, budget and estimated completion timeline. There are models which are suitable for large project, where some focus on lightweight process that allow rapid changes throughout whole software development life cycle.

## 2. PHASES OF SDLC

The SDLC framework consists of a multiple consecutive phases or steps that are to be followed in sequence by software developers and system designers. In each phase of System Development Life Cycle, all the phases are depend on the result of previous phase. The output of the previous phase becomes the input of next phase. The titles of the phases may be varying, depending on the development environment that include planning, analysis, and implementation.

### 1) System conceptualization

In this phase, the System's concept is developed. The thought of a system to be developed, is come into existence. Which kind of system is required?, decides in this phase. Virtually a System is come into existence. The idea of a system, born in this phase. This is base phase behind any new system to be developed. Because the requirement of a system gives the idea of a new system to come into existence. In this phase project's goal is to be determined. Whether to create new project or altering the current project. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals.

The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included.

In this phase all the planning should be done like what to do? What kind of features should be there in software? A kind of prototype would be developed at the planning stage. One more thing is taken into consideration like; the project is feasible in all aspect of the current situation. As a result long term project is planned. The outputs of this phase are these, SDLC description, the Software Configuration Management Plan (SCMP), the Software Quality Assurance Plan (SQAP), the Software Project Management Plan (SPMP) and the associated project or component schedule. The schedule includes a detailed listing of activities for the upcoming requirement stage and high-level estimates of effort for the out stages.

## 2) System Requirements and Benefits Analysis

This phase takes as its input the goals identified in the above phase or system conceptualization. Goal is refined into set of requirements. System requirements are defined in this phase. Requirement of the System is actually there or not, is identify in this phase. Characteristics of System are defined in this phase. All the requirement or need should fulfil by new system. All the requirements to develop a system is analyse this phase. Weather the new system is beneficial over current system or not. All the pros and cons of the new system is analyse in this phase. If the benefits are there of new system is much than current system then the idea to develop new system put forward.

In this phase all the requirement information are to be gathered. Which of them are taken into consideration is decided in this phase. Basis on this information, well-defined functions are created from the defined project goal.

This phase is concerned with establishing what the ideal system has to perform. However it does not determine how the software will be designed or built. Usually, the users are interviewed and a document called the user requirements document is generated. The user requirements document will typically describe the system's functional, interface, performance, data, security, etc requirements as expected by the user. It is used by business analysts to communicate their understanding of the system to the users. The users carefully review this document as this document would serve as the guideline for the system designers in the system design phase. The user acceptance tests are designed in this phase. There are different methods for gathering requirements of both soft and hard methodologies including; interviews, questionnaires, document analysis, observation, throw-away prototypes, use cases and status and dynamic views with users.

These requirements define the major functions of the supposed application and defined the initial data entities. Major functions include critical processes to be managed for i.e. critical inputs, outputs and reports. Each of these definitions is termed a Requirement. Requirements are identified by unique requirements identifiers and at minimum, contain a requirement title and textual description. These requirements are fully described in the primary deliverables for this stage: the Software Requirements Document (SRD) and the Logical Database Description (LDD). The SRD contains complete descriptions of each requirement, including references to external documents, such as Use Cases. The LDD describes the major data entities of the project or component, along with their relationships to other entities and their user base. The identifier associated with each

requirement is also placed into the first version of the Requirements Traceability Matrix (RTM), along with the identifier of each goal from the parent project plan or component iteration plan.

## 3) Project Adoption and Project Scoping

System requirements outputs will be inputs of this phase. After thinking on all the pros and cons of the system requirements, the project is put into developing stage. All the requirements are analyse and decide to develop that project. Weather the current technology is enough to adopt a new project. Which is new technology to adopt to cope with new project? What is the future scope of the project? All these things must be keep in mind. Feasibility study of the system is done over here. There are four types of feasibility study. Technical feasibility, economical feasibility, operational feasibility, and organizational feasibility.

## 4) System Design

After getting positive result from previous phase the System is designed in detail. Outputs of the system requirements will be inputs of this phase.

Project is designed in this phase. Whole project/system design is developed. All operations and features are described in detail, which include technical specification. UML is used in this phase when required. Process diagrams are prepared to facilitate the SDLC process with required documentation. In short, blue print of the whole system is develop in this phase. How the system will work, is decided.

System design is the phase where system engineers analyze and understand the business of the proposed system by studying the user requirements document. They plan our possibilities and techniques by which the user requirements can be implemented. If any of the requirements are not feasible, the user is informed about this issue. A resolution is found and the user requirements document is edited accordingly. Establishes the expectations for software functionality and identifies which system requirements the software affects. Requirements analysis includes determining interaction needed with other applications and databases, performance requirements, user interface requirements, and so on.

Design elements describe the desired software features in detail. The Software Design Document (SDD) contains the functional or dynamic design elements, such as business rules, business process diagrams. Physical Database Description (PDD) contains the static or structural design elements such as the entity relationship diagram, the access control matrix.

The software specification document which serves as a blueprint for the System development phase is generated in this phase. This document contains the general system organization, menu structures, data structure etc. It also hold system scenario, sample windows, reports for the better understanding. Other technical documents like entity diagram, data dictionary will also be produced in this phase. When the SDD and PDD are finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement.

## 5) Specification of software requirements

All the technical specification is describe in this phase. Which tool are required to develop this project? The need for the

specific platform and technology is also decide in this phase of the develop. Specify also, if there is any need of other tools or technology to build the project.

## 6) Architectural design

The design of the Whole System is precise briefly in this phase. Whole system is divided in different modules or units. And documentation is prepared in this unit according to completion of the previous phase. Functionality of each unit is design in this phase. Determines the software framework of a system to meet the specific requirements. This design defines the major components and the interaction of those components, but it does not define the structure of each component. The external interfaces and tools used in the project can be determined by the designer. This phase of the design of computer architecture and software architecture can also be referred to as high level design. The baseline in selecting the architecture is that in this phase list of modules, brief functionality of each module, their interface relationship, dependencies, database tables, architecture diagrams, technology details etc are prepared.

## 7) Detailed design

In this phase all the unit are designed in detail. As functionality of each unit is designed in previous phase. How this unit works, is design in this phase? Which kind of functions should be there and functionality provided by that functions are decided in this phase. Characteristics of all the units are decided in this phase. The designed system is broken up into smaller units or modules and each of them is explained so that the programmer can start coding directly. The low level design document or program specification will contain a detailed functional logic of the module or pseudo code. Database tables with all elements, including their type and size are prepared in this phase. All dependency issues, error message list are prepared in this phase. Complete inputs and outputs are decided for a module.

## 8) Unit development

All the units are actually design in this phase. This is the coding phase. Coding is being done in this unit. Units are developed under the technology, which is decided in previous phase. This is the actual phase where project is converted into reality. This development stage takes as its inputs the design elements describes in the approved SDD and PDD. For each design element, a set of one or more software objects are produced. Software objects include but are not limited to menus, dialogs, data entry forms, data reporting formats, and specialized procedures and functions. Appropriate test cases are developed for each set of functionally related software objects and an online help system is developed to guide users in their interactions with the software.

The RTM is updated to show that each developed object is linked to a specific design element and that each developed object has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an Implementation Map (IMP) that identifies the primary code entry points for all major system functions, a Software Test Plan (STP) that describes the test cases to be used to validate

the correctness and completeness of the software, an updated RTM.

## 9) Unit testing

In computer programming, unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual function or procedure. Unit tests are created by programmers or occasionally by white box testers. The purpose is to verify the internal logic code by testing every possible branch within the function, also known as test coverage. Static analysis tools are used to facilitate in this process, where variations of input data are passed to the function to test every possible case of execution. It is main type of testing.

## 10) Software integration & testing

Integrated module is tested for their interoperability and functionality. In integration testing the separate modules will be tested together to expose faults in the interfaces and in the interaction between integrated components. Testing is usually black box as the code is not directly checked for errors. During the integration and test phase, the software objects, online help and test data are migrated from the development environment to a separate test environment. At this point, all the cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability.

During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Deployment Plan (DP). The outputs of the integration and test phase include an integrated set of software, an online help system, an updated Implementation Map (IMP if necessary), a Deployment Plan (DP) that describes reference data and production users, an acceptance plan which contains the final suite of test cases.

## 11) System integration & testing

System testing will compare the system specification against the actual system. After the integration test is completed, the next test level is the system test. System testing check if the integrated product meets the desired requirements. It is done because the customer who has ordered and paid for the system and the end user who will use the system may be different group of people or organizations with their own specific interests and requirements of the system. So software testing is done against technical specifications. And the system test, looks at the system from the perspective of the customer and the future user. The tester validate if the requirements are completely and appropriately met or not. Many functions and system characteristics result from the interaction of all system components, so they are only visible on the level of the entire system and can only be observed and tested there.

## 12) Installation at site

After whole system is integrated and tested, the project is installed at customer's place. All the applications which are required to run this project is also installed at the customer's place.

### 13) Site testing and acceptance

After installing the project at its customer's place, the whole project is tested again. If it works properly without any problem then it is accepted by the customer.

### 14) Training and documentation

How the project work and how to operate the project, all this information is provided to the end user in training. Documentation of whole project is also provided to the end user for their future need. In the documentation all the required information related to the project is given, which help to the end user.

### 15) Implementation

This is the phase where the developed software is actually used by the end user. Whole the system is used for the task for which purpose that project is develop.

Actual use of the project is occurred in this phase of the SDLC life cycle model.

### 16) Maintenance

This phase is very important in SDLC model. Because this phase can become a project in and of itself. Future software upgrades, bug fixes, and regular maintenance are addressed during this phase. This phase may or may not have a well defined end state, as if user wants to see some changes in future. So it's keeping on maintaining the software.

## 3. CONCLUSION

Researchers have tried to explain all the phases of Software Development Life Cycle in detail. This will help programmer

to develop any software very smoothly and easily. This paper would lead programmer to do their job without any problem. This will facilitate the programmer to build the software stepwise, so confusion can be avoid in any phase of development.

## 4. REFERENCES

- [1] [www.en.wikipedia.org/wiki/Systems\\_development\\_life-cycle](http://www.en.wikipedia.org/wiki/Systems_development_life-cycle)
- [2] [www.en.wikipedia.org/wiki/SDLC](http://www.en.wikipedia.org/wiki/SDLC)
- [3] [www.waterfall-model.com/sdlc/](http://www.waterfall-model.com/sdlc/)
- [4] [www.condor.depaul.edu/jpetlick/extra/394/Session2.ppt](http://www.condor.depaul.edu/jpetlick/extra/394/Session2.ppt)
- [5] [www.searchsoftwarequality.techtarget.com/.../systems-development](http://www.searchsoftwarequality.techtarget.com/.../systems-development)
- [6] <http://searchsoftwarequality.techtarget.com/answer/Software-development-life-cycle-phases-iterations-explained-step-by-step>
- [7] [www.allinterview.com](http://www.allinterview.com) › Categories › Software
- [8] [www.vtlglobal.com/software-development-life-cycle.html](http://www.vtlglobal.com/software-development-life-cycle.html)
- [9] [www.e-digg.com/services/software/sdlc-life-cycle.html](http://www.e-digg.com/services/software/sdlc-life-cycle.html) - United States
- [10] [www.shazsoftware.com/software-development-life-cycle.html](http://www.shazsoftware.com/software-development-life-cycle.html)