

INTERSHIP REPORT  
**SOFTWARE DEVELOPMENT – TECHNICAL  
INTERN**

BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE

BY

**S PRIYA DHARSHINI**

ROLL NO: 22BSCS243

UNDER SUPERVISION OF

**M PRASANNA**

CHIEF OPERATIONS OFFICER, STRARTS COURSES

CHENNAI, INDIA

DURATION: 03 JUN 2023 to 17 JUN 2023



**SRI KRISHNA ADITHYA  
COLLEGE OF ARTS & SCIENCE**

Recognized with 2(f) & 12(B) status by UGC | Affiliated to Bharathiar University  
Kovaipudur, Coimbatore - 641042



## DEPARTMENT OF COMPUTER SCIENCE

JUNE 2023

### CERTIFICATE OF COMPLETION

This is to certify that **S PRIYA DHARSHINI** [22BSCS243] from SRI KRISHNA ADITHYA COLLEGE OF ARTS AND SCIENCE, Department of Computer Science completed the internship training in Software Development Lifecycle/Agile Methodologies in STRARTS Courses from 03 JUN 2023 to 17 JUN 2023.

Dr THILAGAVATHI, Ph.D.,

Assistant Professor, Computer Science

Sri Krishna Adithya College of Arts & Science,

Coimbatore.

Dr CHITRA, M.C.A, M.Phil., Ph.D.,

Head of the Department, Computer Science,

Sri Krishna Adithya College of Arts & Science,

Coimbatore.

## ACKNOWLEDGMENT

First, I would like to thank Mr. PRASANNA, CHIEF OPERATIONS OFFICER OF STRARTS COURSES, for giving me the opportunity to do an internship with this organization. I also would like to thank STRARTS COURSES for the enjoyable environment created. It is indeed with immense pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I take this opportunity to express my profound gratitude and whole thanks to our Principal Dr PALANIAMMAL S, M.Sc., M.Phil., Ph.D., Sri Krishna Adithya College of Arts & Science for reinforcing my effort with support and guidance.

I would like to thank Head of the Department Dr CHITRA K, M.C.A, M.Phil., Ph.D., Assistant Professor & HOD, Department of Computer Science for her guidance and throughout my internship.

I would like to thank my class advisor Dr THILAGAVATHI, Ph.D., Assistant Professor, Department of Computer Science for her support and advice to complete my internship in above mentioned organization.

## **SOFTWARE DEVELOPMENT LIFECYCLE AND AGILE METHODOLOGIES**

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.
- ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

A systematic approach that generates a structure for the developer to design, create and deliver high-quality software based on customer requirements and needs. The primary goal of the SDLC process is to produce cost-efficient and high-quality products. The process comprises a detailed plan that describes how to develop, maintain, and replace the software.

### **Stage 1: Planning and Requirement Analysis**

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

### **Stage 2: Defining Requirements**

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

### **Stage 3: Designing the Product Architecture**

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third-party modules (if any).

## **Stage 4: Building or Developing the Product**

In this stage of SDLC the actual development starts, and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Java, and Python are used for coding. The programming language is chosen with respect to the type of software being developed.

## **Stage 5: Testing the Product**

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

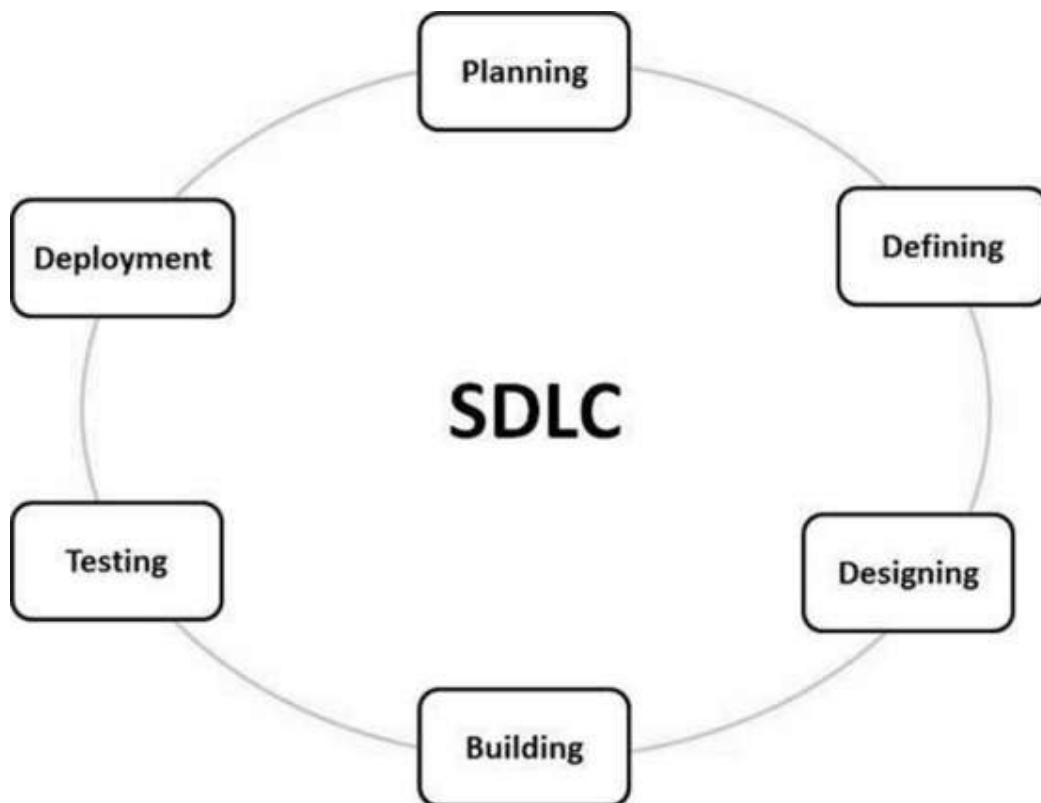
The detailed testing process is as follows:

1. Unit testing - Testing done by the developer to confirm that the basic functionality of the product is working.
2. Functional testing - Deep testing of the functionalities with all successful, failure and regression-based scenarios.
3. Integration testing - Testing the integration components if there is any integration with third party systems.
4. User Acceptance testing - User or the customer performing the testing and confirming if the software matches the requirement.

## **Stage 6: Deployment in the Market and Maintenance**

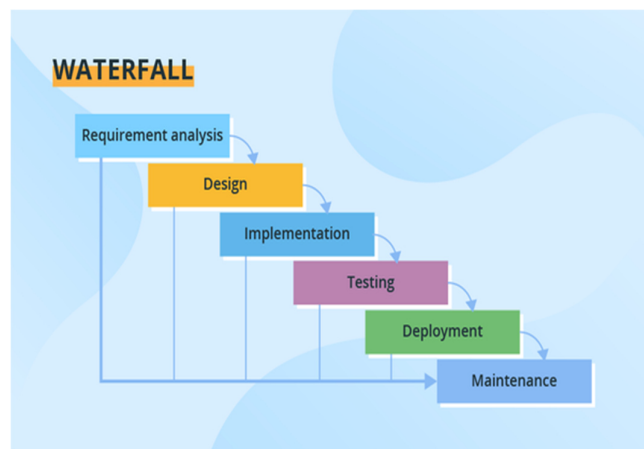
Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

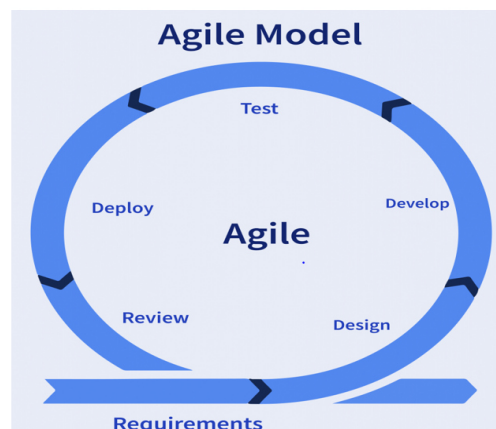


# Predominant Models of SDLC (Software Development Life Cycle)

**Waterfall Model:** This SDLC model is the oldest and most forthright. We finish with one phase and then start with the next, with the help of this methodology. Why the name waterfall? Because each of the phases in this model has its own mini-plan and each stage waterfalls into the next. A drawback that holds back this model is that even the minute details left incomplete can hold an entire process.

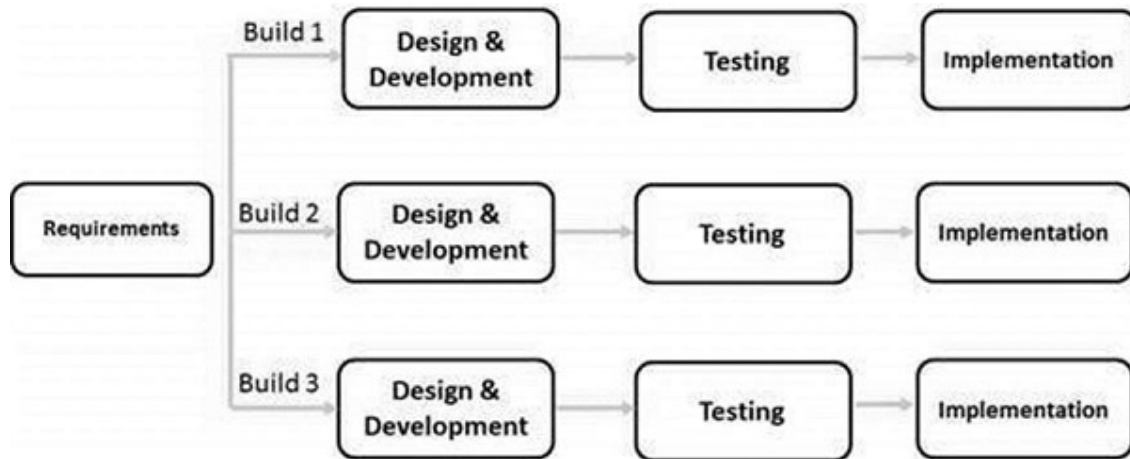


**Agile Model:** Agile is the new normal; It is one of the most utilized models, as it approaches software development in incremental but rapid cycles, commonly referred to as “sprints”. With recent changes in scope and direction being implemented in each sprint, the project can be completed quickly with higher flexibility.

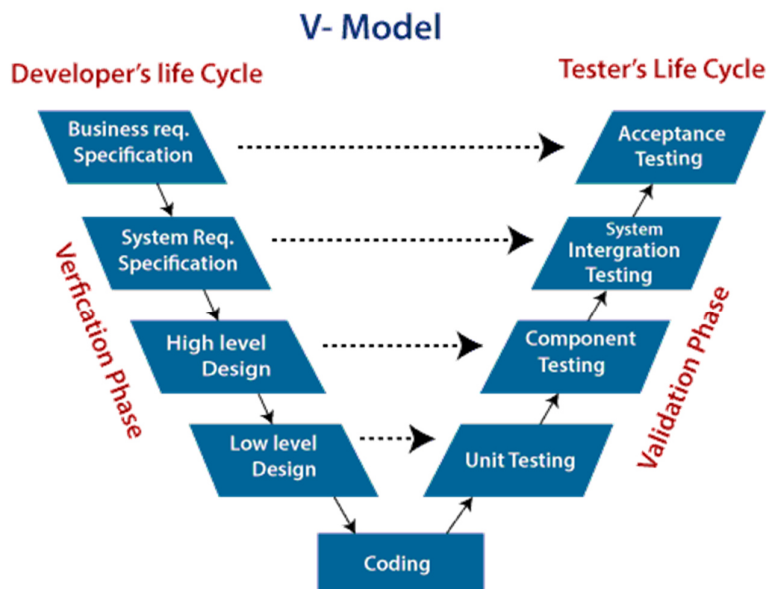




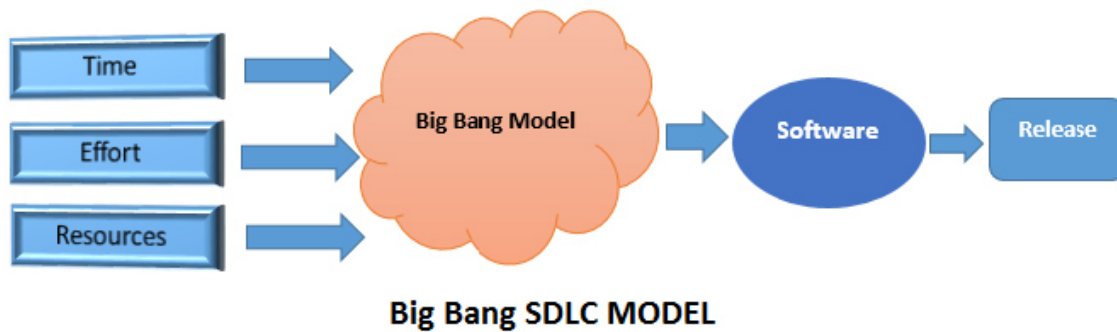
**Iterative Model:** This SDLC model stresses on repetition. Developers create a version rapidly for relatively less cost, then test and improve it through successive versions. One big disadvantage of this model is that if left unchecked, it can eat up resources fast.



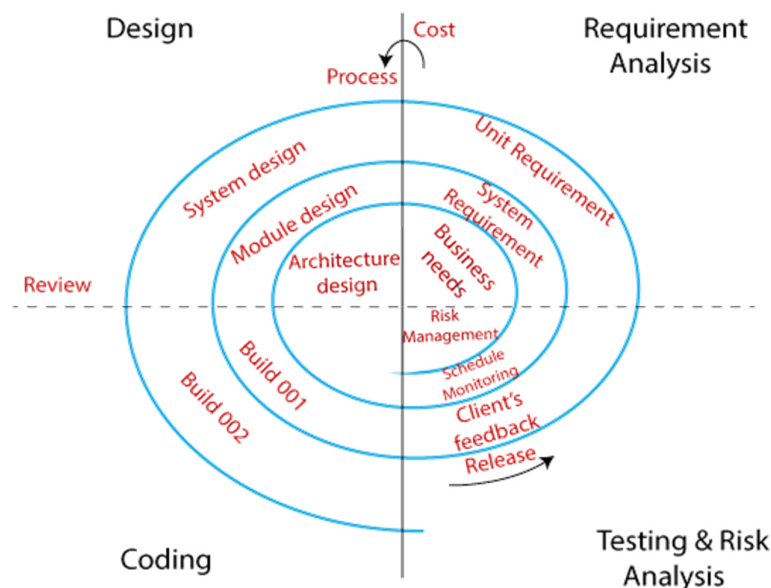
**V-Shaped Model:** This model can be considered as an extension of the waterfall model, as it includes tests at each stage of development. Just like the case with waterfall, this process can run into obstructions.



**Big Bang Model:** This SDLC model is considered best for small projects as it throws most of its resources at development. It lacks the detailed requirements definition stage when compared to the other methods.



**Spiral Model:** One of the most flexible of the SDLC models is the spiral model. It resembles the iterative model in its emphasis on repetition. Even this model goes through the planning, design, build and test phases repeatedly, with gradual improvements at each stage.



This company STRARTS Courses works on agile methodologies to design and deliver products with better functionalities and efficiency.

One of the easiest and most successful ways to transform a vision for a company requirement into software solutions is to use the Agile software development approach. Continuous planning, learning, improvement, team collaboration, evolutionary development, and early delivery are all terms used to define agile software development methodologies. It increases adaptability in the face of change.

The four essential values of agile software development are highlighted.

- Interactions between individuals and groups around processes and tools
- Working software models and thorough documentation.
- Collaboration with customers is preferred above contract negotiations.
- Adapting to change in accordance with a strategy

With the above values in place, agile methodology steps include:

- Sprints and iterations – A stipulated period or time window
- A product backlog creation - This is a list that contains all the information/features needed to create the product.
- The top Features/Functionality from the Product backlog are chosen and translated into Sprint backlogs during each Sprint.
- The team works on the sprint backlog that has been established.
- The team double-checks the everyday job.
- The team delivers the product functionality after the sprint.

Agile is also a process that enables continuous improvement of a product with quality feedbacks, reviews along the development life cycle.