Software Development Life cycle:

Methods to make a software product through its stages within a software organization

It aims to produce high quality software that exceeds customer expectations, reaches completion within times and cost estimates

Detailed plan of how to develop, maintain, replace and alter or enhance specific software

**Needs:**

Without using an exact life cycle model, Product would not be in a systematic and disciplined manner

The members of a software development team must agree on when to do what and when not to do it when developing software. Otherwise, chaos and project failure would result

Without software life cycle models, it becomes tough for software project managers to monitor the progress of the project. Because software life cycle model describes entry and exit criteria for each phase



Fig source: <https://www.tutorialspoint.com/sdlc/sdlc_overview.html>

Stage 1: Planning and Requirement Analysis

It is the most important stage of sdlc. Performed by senior members of the team with customer inputs, market surveys, the sales department and domain experts in industry

Based on this information, the basic project approach is planned and a feasibility study is conducted on the economics, operations, and technical aspects of the product

Stage 2: Defining Requirements

Once first stage is done, then clearly define and document the product requirements and get them approved from the customer or market analysts.

This is done through SRS (Software Requirement Specification) document. It consists of all product requirements to be designed and developed during the project life cycle

Stage 3: Designing the product architecture

It is about knowledge of requirements, analysis, and design of software project. This phase is the product of the last two, like inputs from the customer and requirement gathering

Stage 4: Building or Developing the product

In this phase, the actual development starts and product is built. The programming code is generated as per DDS (design document specification)

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers etc. are used to generate the code

Step 5: Testing the product

Subset of all stages as in the modern SDLC models, testing activities are mostly involved in all stages all the stages of SDLC. 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

In other words,

After code is done, it is tested against the requirements to make sure that solving needs addressed and gathered during the requirements stage

Stage 6: Deployment

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.

Product may first be released in a limited segment and tested in the real business environment

After the software is deployed, then its maintenance begins

Agile Software Development life cycle

It is the combination of both Iterative and incremental process models. It focuses on process adaptability and customer satisfaction by rapid delivery of working software product

Agile SDLC breaks down the product into small incremental builds. These are provided into iterations

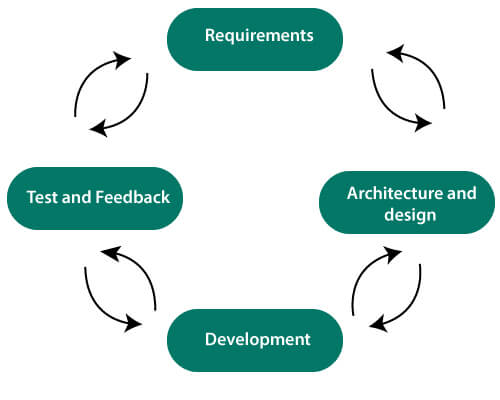


Fig Source: <https://www.javatpoint.com/agile-sdlc>

The customer can see the results and decide whether he/she is satisfied A major advantage of agile SDLC is its flexibility. Its disadvantage is that there are no defined requirements, making it difficult to estimate the resources and development costs

* Requirement gathering and analysis:

In this phase defining the requirements, explaining business opportunities and plan the time and effort needed to build the project. Based on this information can evaluate technical and economic feasibility

* Design the requirements:

Work with stakeholders to define requirements. Using UML, user flow diagram to show the work of new features and show how it will apply to your existing system

* Construction/Iteration:

The designers and developers start working on their project. The aims of designers and developers deploy the working product within the estimated time. The product will go into various stages of improvement, so it includes simple, minimal functionality.

* Deployment:

In this phase, the team issues a product for the user’s work environment

* Testing:

In this phase, Quality Assurance team examines the product performance and look for the bug

* Feedback:

After releasing product, in this step team receives feedback about the product and how it works

Advantages:

* Project divided into short and clear iterations
* Flexible changes
* Minimizes of risk
* Quick release of version

Disadvantages:

* Development team should be highly professional
* New requirements can cause conflicts
* With further modifications, expected time may change
* Difficult to estimate final coast of project due to constant iteration
* Defined requirement is absent

DeVops

combination of the terms *development* and *operations*, meant to represent a collaborative or shared approach to the tasks performed by a company's application development and IT operations teams

While DevOps is not a technology, DevOps environments generally apply common methodologies

methodology meant to improve work throughout the software development lifecycle. You can visualize a DeVops process as an infinite loop, comprising these steps: plan, code, build, test, release, deploy, operate, monitor and -- through feedback -- plan, which resets the loop

an IT team writes software that perfectly meets user requirements, deploys without any wasted time and runs optimally on the first try. Organizations use a combination of culture and technology to pursue this goal.

* fewer silos and increased communications between IT groups;
* faster time to market for software;
* rapid improvement based on feedback;
* less downtime;
* improvement to the entire software delivery pipeline through builds, validations and deployment;
* less menial work, thanks to automation;