What is the software development life cycle (SDLC)?

The software development life cycle is a process of planning, creating, testing, and deploying information systems across hardware and software.

[Demo DevOps](https://www.servicenow.com/lpdem/demonow-devops.html)

Software development is an iterative process that is followed for a software project that consists of several phases for building and running software applications. SDLC helps with the measurement and improvement of a process, which allows an analysis of software development each step of the way.

How SDLC works

SDLC simply outlines each task that is required to create and deploy a software application—this prevents the reduction of waste and increases efficiency along the way. Monitoring that is a part of SDLC helps companies ensure that everything is timely, budgets are adhered to, and the software continues to be a feasible investment. SDLC is often described as leveraging Agile or Waterfall approaches and many organizations use a hybrid of both.

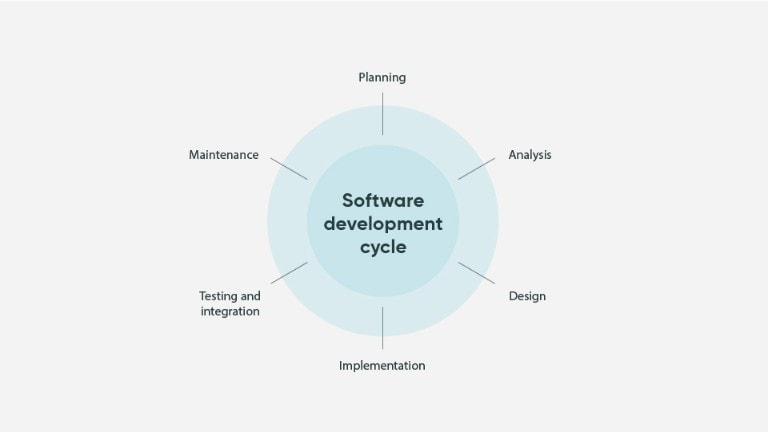
Defining the software development life cycle

1. Identify the Current Problems

ITSM is a strategic approach to IT management, with a focus on delivering value to customers. ITSM clearly defines the roles and responsibilities of every individual and department with regard to IT services. It allows for increased productivity, lower costs, and improved end-user satisfaction.

2. Planning

Leaders of the project evaluate the terms of the project, including the calculation of labor and material costs, to create a timetable with goals. Planning must also include areas of feedback from stakeholders or anybody who is going to benefit from the applications. The scope of the project should be clearly defined, the purpose of the application outlined, and the boundaries that are needed to keep the project from expanding beyond scope or shifting.



3. Define Requirements

The next step is to define and document requirements and seek stakeholder approval. Define what the application is meant to do, any features that would be included, and roadblocks along the way. Resources would also need to be identified and built into the project in order to define requirements.

4. Design and Prototyping

SDLC requires a designing step that models how the application will work and aspects of the design. Some of the aspects can include:

* UI: How customers will interact with the software and how the software is meant to respond to certain inputs.
* Programming: The programming language that will be used, as well as how the software will sofe problems and perform tasks.
* Security: The certain measures that will be taken to ensure that the application is secured. This includes SSL encryption, password protection, and secure data storage.
* Communications: Define how the application will communicate with other assets like a central server.
* Architecture: Includes industry practices, any templates, overall design, and specific programming languages.
* Platforms: Outlines the platform that will host the software, like Apple, Windows, Android, or Linux.

After the design has been defined, a prototype of an early version of the software can be created to demonstrate a basic idea of how an application will look, how it will respond, and what it is capable of doing. This is the phase where programmers receive feedback from stakeholders in order to approve the application—prototyping is much less expensive than making changes in the development phase.

5. Software development

This is the phase of SDLC where the program itself is written out, either using a single developer or a large team each working on different parts of the development. SDLC can anticipate issues in the software development process that can hold up production, like waiting for test results or compiling code.

6. Testing

Applications must be tested continuously to ensure that they are going to run well together, as software development is often broken down into smaller projects completed by separate individuals and teams. Ensure that each function runs as it should and that each part of the application is interacting well with other parts. This reduces the number of bugs that users can encounter when using the application, and it leads to a higher usage rate and better user satisfaction.

7. Deployment

An application is deployed once testing is completed, which makes it available to users. This step of the process can be manual or automated, depending on the complexity and needs of the application.

8. Operations and Maintenance

Once the application has been deployed and is being used, the final phase discovers bugs that slipped through the cracks during testing and resolves them—this can start its own iterative process.

SDLC Methodologies

There are several different approaches to software development, each different enough that an organization can find their needs fulfilled with one of them.

Waterfall

Just as a waterfall flows in a linear fashion, waterfall SDLC is a linear model that has the development of the software start from the beginning and move through each step of the process—but, the next step cannot start until the prior step has been completely finished. Waterfall SDLC helps a company analyze continuity and feasibility of each step of the process, which can help eliminate bottlenecks or silos.

Agile

The agile methodology focuses strongly on user input and experience, which can solve many issues that arose from older applications that were more cumbersome to use. The software, as it moves through the agile process, is very responsive to feedback and works to release software in quicker cycles in order to adhere to a changing and rapid market.

Iterative

Software developers quickly create an initial version of the software, which is then improved upon in small iterations. This is an approach commonly used in larger applications that can help get the application up and running to meet a business need more quickly.

DevOps

Very similar to agile, DevOps looks to improve the usability and relevance of an application that has been developed by gathering feedback from the software users during the design and implementation phases—all while using active collaboration and communication between members of a team that has been assembled to cover all aspects of a product. This team will include representation for information technology operational roles as well as development, testing and security.

Spiral

Spiral uses the iterative approach of other models in conjunction with sequential processes as seen in waterfall. This allows for incremental releases and refinement through each step of the spiral as the development goes through each step of the process repeatedly.

V-model

Also known as the verification and validation model, the V-model approaches the steps sequentially, but in a “v” shape—it is an extension of the waterfall approach that includes testing that is directly associated with each phase of the development rather than testing in a single phase.

Big bang

There is very little planning and very few processes involved in the big bang model. The process begins with the necessary funding and resources, then output software that hopefully follows customer requirements. This model is usually used for very small projects with very small teams that don’t require as much formal structure and process.

Benefits of the SDLC

SDLC provides an upfront, centralized goal for everyone to agree with and understand. There is a clear plan for development, including resources and associated costs, and roles can be assigned out to ensure proper execution of the plan.

Best Practices of Software Development

Source Control

Source control keeps all of the code in a single location in order to secure the working code. This can be a physical location or a virtual location wherein users can login to an encrypted cloud environment.

Continuous Integration

Make sure that each component of the asset is consistently compatible throughout the life cycle. Continuous integration ensures that all team members avoid conflicts and duplicates by using similar programming languages and libraries.

SDLC Management Systems

Add transparency to systems through each phase of the project, and throughout the project as a whole. SDLC management systems control each step of the way while adding analytics, work management systems, and bug-tracking that can improve parts of the lice cycle that aren’t running effectively.

The software development life cycle is an excellent process that can help teams through the process of developing and deploying software, including necessary fixes, by utilizing best practices and methodologies.