# **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.

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.

## **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.