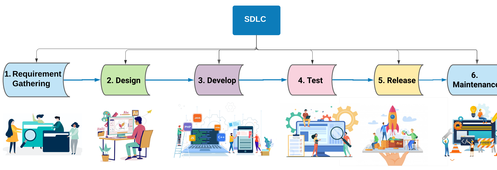
Software Development Life Cycle (SDLC)

Interview question from this topic:

- What is **SDLC** ?

Try to fill out "SDLC Summary" sheet while reading this article.

Click here to get the sheet



Step 1: Requirement Gathering -> What does the client want?

This is the **first step** to build a software. In this step, the Product Owner (**PO**) and **the business team collects** the **requirements from the client** who has the idea of the application.

SDLC or the **Software Development Life Cycle** refers to well-structured **process** clearly defined **for building** high-quality **software**. Any IT company follows these steps to build applicaiton.

The SDLC includes 6 systematic phases/steps :

1. Requirement Gathering
2. Designing
3. Coding / Buidling / Impelmenting
4. Testing / Quality Assurance
5. Deploying / Releasing / Production
6. Maintaining

Requirement **: Requirement is the** description of features or functionalities of the target software. Expectation of the customer. There are Functional Requesrements and Non-Functional Requirements.

The Business team **analyzes the requiremetnts** and **plans** the **costs** of developing a product, **capacity** of **team members**, **project schedule**, and resource allocation.

After analyzing and planning the software, the **business team creates several documents**. One of the important documents is **the** SRS - **Software Requirement Specification document.**

Click here to see a smaple SRS document for a Bank management software

SRS document includes all the **functional** and **non-functional requirements**(a set of detailed descriptions) for the software they plan to build. The SRS also **describes** the business team's analysis result for the app and any hardware and software requirements.

Note: The other documents are not critical to be learnt. Based on the companies and their needs, the business teams may also prepare several other documents like Market requirement doc (MR), Functoinal requirment(FR) doc, Non-functional requirements doc(NFR), etc.

Functional requirements are the goals of the application or product. They describe **how the application** will **function**.

For Example:I want online store software. Users add items in the shopping cart and order products. The software can pay online, information about the company, and feedback section.

Non-functional requirements are the general characteristics that affect user experience. These are requirements for **data integrity**, **security**, **speed**, **capacity**, **stress**, **installation** and **licensing policy**.

For example: The project homepage should load in 2 seconds. Users should be able to download the software to both MacOS and Windows.

To write a good SRS doc, The business team has to follow a ***SMART*** criteria. Objective of SMART should be Specific, Measurable, Achievable, Relevant, and Tesable.

Step 2: Designing -> How will we get what we want?

This is the second step / phase of SDLC. The design phase of the SDLC starts by **turning** the software requirement Specifications (**SRS**) **into** a design plan called the Software Design Specification (**SDS**).

Click here to see a sample SDS document for bank software

Another sample SDS document of bank application

SDS document includes the detail for the overall software such as:

- User interfaces
- System interfaces
- Network and network requirements
- Databases

Also, **designers** may use designing tools to **design** a **prototype of the software** according to the SRS. The client then review this plan and offer feedback and suggestions.

Requirement: user should able to select peyment options.

Which design is the best ? Radio button ? or Drop-down?

Step 3 : Coding / Building / Implementing -> Start to create what we want.

Coding or developing is the third step of SDLC. At this stage, the **actual development starts**. **Developers** work to build the software according to the SRS & SDS documents and feedback.

- Codes are divided into small units
- Developers review each others' code
- Developer lead approves the codes
- This is the longest phase of the SDLC process

Step 4 Testing -> Did we get what the client wants?

Testing is the fourth step of SDLC. In this stage, **testers test the software for identifying bugs** and defects. Any bugs or defects need to be tracked, fixed, and retested.

The software's **functionality**, **performance**, and **security** levels **are being tested manually and automatically.** Developers fix bugs until the product meets the original specifications.

Testers **Perform software testings** based on functional and non-functional requirements.

Analyze the requirement and **plan** how to test

Write test case **documents** and prepare test data

Identify bugs & write bug reprots

Test reports are documented

Step 5 Deploying / Release / Production-> Let the end-users start using what we got.

After testing step, the software will be **release to the world so target audience can use it**.

- Developers, testers, business team members are all together release the app
- Codes / new features are deployed to production
- The software will be practical used by end users

Step 6: Maintaining -> Let's get this closer to what the client wants.

Maintaining is a phase where the software is **monitored** and **maintained** to **ensure** its **functionality** and **quality** are not deteriorating in the real production environment. If any bug is not found in the testing step, **end-users may discover bugs** while using the software. The developers must fix any **bugs** found in **production** (we call this **hotfix**). Also, make sure the fix does not introduce more bugs.

If the client **provides new requirements** or **feedback** to improve the software. ***For the new requirements, a new Software Development Cycle is launched.***In other words, all the team members will start from the requirement gathering step again to develop the new features. That is why software development is a life cycle.