The Software Development Life Cycle (SDLC):

The software development life cycle is a process that development teams use to create awesome software that's top-notch in terms of quality, cost-effectiveness, and time efficiency. The main goal is to minimize risks long before deciding how to launch your product and ensure the software meets the customer's expectations during and after production.

This process is about creating a detailed plan to guide the development of the product and then breaking down the development process into smaller modules that can be assigned, completed, and measured to make the whole thing more manageable.

Here are some specific benefits of using SDLC for the product team:

- Increased visibility of the development process for all stakeholders involved
- More efficient estimation, planning, and scheduling
- Improved risk management and cost estimation
- A systematic approach to delivering software that meets customer expectations and improves satisfaction

The 7 Phases of the Software Development Life Cycle

The SDLC process will look a little different for every team and product. However, these are the stages that most SDLC frameworks have in common:

1. Planning & Analysis

The first phase of the SDLC is the project planning stage where you are gathering business requirements from your client or stakeholders.

2. Define Requirements

This phase is critical for converting the information gathered during the planning and analysis phase into clear requirements for the development team. This process guides the development of several important documents: a software requirement specification (SRS) or product specification,

3. Design

The design phase is the original plan and vision are elaborated into a software design document (SDD) that includes the system design, programming language, templates, platform to use, and application security measures. This is also where you can flowchart how the software responds to user actions.

4. Development

The actual development phase is where the development team members divide the project into software modules and turn the software requirement into code that makes the product.
This SDLC phase can take quite a lot of time and specialized development tools. It's important to have a set timeline and milestones so the software developers understand the expectations and you can keep track of the progress in this stage.

5. Testing

Before getting the software product out the door to the production environment, it's important to have your quality assurance team perform validation testing to make sure it is functioning properly and does what it's meant to do. The testing process can also help hash out any major user experience issues and security issues.

In some cases, software testing can be done in a simulated environment. Other simpler tests can also be automated.

The types of testing to do in this phase:

Performance testing: Assesses the software's speed and scalability under different conditions
Functional testing: Verifies that the software meets the requirements
Security testing: Identifies potential vulnerabilities and weaknesses
Unit-testing: Tests individual units or components of the software
Usability testing: Evaluates the software's user interface and overall user experience
Acceptance **testing:** Also termed end-user testing, beta testing, application testing, or field testing, this is the final testing stage to test if the software product delivers on what it promises

6. Deployment

During the deployment phase, your final product is delivered to your intended user. You can automate this process and schedule your deployment depending on the type.

7. Maintenance

The maintenance phase is the final stage of the SDLC if you're following the waterfall structure of the software development process. However, the industry is moving towards a more agile software development approach where maintenance is only a stage for further improvement.

SDLC and Security

It should come as no surprise that security is an increasing concern in the software world. Building security into a software product is a project in and of itself, so these operations are typically integrated into the software development life cycle.

How can you integrate security into the SDLC?

SDLC integrates security via DevSecOps, which is not an isolated stage but a continuous process.

DevSecOps, an extension of DevOps, incorporates security checks at every SDLC phase. Activities include code review, architecture analysis, penetration testing, and automated detection. Tools are integrated into IDEs, code repositories, and build servers.

How to Integrate DevSecOps into SDLC?

1. Planning & Requirement Analysis.
2. Architectural Design.
3. Software Development & Testing.
4. Deployment.
5. Maintenance.

Common SDLC Models

In software development, there are various frameworks, or "models," of the Software Development Lifecycle (SDLC), which arrange the development process in different ways. These models help organizations implement SDLC in an organized way. Here are some of the most commonly used software life cycle models.

1. Agile Model.
2. Waterfall Model.
3. Iterative Model.
4. Spiral Model.
5. Big Bang Model.