- A Software Development Life Cycle (SDLC) is a framework that defines the process used by organizations to build an application from its inception to its decommission. Over the years, multiple standard SDLC models have been proposed (Waterfall, Iterative, Agile, etc.) and used in various ways to fit individual circumstances. It is, however, safe to say that in general, SDLCs include the following phases:
- Planning and requirements.
- Architecture and design.
- Coding.
- Testing and results.
- Release and maintenance.

- In the past, it was common practice to perform security-related activities only as part of testing. This after-the-fact technique usually resulted in a high number of issues discovered too late (or not discovered at all). It is a far better practice to integrate activities across the SDLC to help discover and reduce vulnerabilities early, effectively building security in.

- It is in this spirit that the concept of Secure SDLC arises. A Secure SDLC process ensures that security assurance activities such as penetration testing, code review, and architecture analysis are an integral part of the development effort. The primary advantages of pursuing a Secure SDLC approach are:
- More secure software as security is a *continuous* concern.
- Awareness of security considerations by stakeholders.
- Early detection of flaws in the system.
- Cost reduction as a result of early detection and resolution of issues.
- Overall reduction of intrinsic business risks for the organization.

- Generally speaking, a Secure SDLC is set up by adding security-related activities to an existing development process. For example, writing security requirements alongside the collection of functional requirements, or performing an architecture risk analysis during the design phase of the SDLC.