**Software Development Life Cycle (SDLC):**

The SDLC, or Software Development Life Cycle, is a process used by software industries for the purposes of designing, developing and testing software. Ultimately, the aim of the process is to produce high quality software that meets (or exceeds) the expectations of the client while still keeping within the time and cost estimates.

As the SDLC is a methodology that defines the steps of a software development project, it consists of multiple phases as well as types (models) depending on the client’s situation, which is further elaborated below.

**Phases of the SDLC:**

1. Requirements Gathering & Analysis

* In the first phase, the project’s approach, deliverables and anticipated final outcome are all determined based on the business requirements provided by the client, such as who will use the software and how they will do so.

1. Software Design

* In the second phase, several (architectural, user interface, database, etc) designs of the software are produced based on the requirements gathered in the requirements gathering and analysis phase. Each design is then reviewed before the best design is selected.

1. Coding & Implementation

* In the third phase, the chosen design documents from the software design phase are used to implement the design and to produce the code needed for the software.

1. Testing

* In the fourth phase, the code is thoroughly tested based on the client's requirements to ensure that it meets the specifications and runs well.

1. Deployment

* In the fifth phase, the finished software is then delivered to the client and may also involve some form of training on how to operate the software as well.

**Software Development Models:**

1. Waterfall Model

* The waterfall model follows a linear sequential flow where progress is seen as flowing downwards (similar to a waterfall) through the multiple phases of software implementation. This essentially means that any subsequent phase in the software development process only begins after the previous phase is completed.
* This model is widely used for software development, when the client has made very clear documented requirements about the software from the start.

1. Evolutionary Prototyping Model

* The evolutionary prototyping model consists of creating multiple prototypes of the software program throughout the development process based on the client’s feedback. This is done so as to better visualize the different components of the software in order to bridge any misunderstandings between the client’s requirements and the development team.
* This model is used when there is increased involvement by the client during the development of the software.

1. Agile Model

* The agile model is based on iterative and incremental development, where the requirements and solutions of the project constantly evolve through collaborations between the cross-functional teams. This often requires face-to-face communication and continuous input from the client (or his representative), thus leaving no room for guesswork.
* This model is used when the client needs to have the software functional and ready in the least amount of time (usually less than 3 weeks) and the requirements are not that clear enough.

**References:**

* <https://www.tutorialspoint.com/sdlc/sdlc_overview.htm>
* <http://study.com/academy/lesson/what-is-a-software-development-life-cycle-definition-examples.html>
* <https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/>