**Software Development Life Cycle(SDLC)**

Software development life cycle is a framework defining tasks performed at each step in the software development process. It is also a structure followed by a development team within the software organization. It consists of a detailed plan describing how to develop, maintain and replace specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. There are six phases in every Software development life cycle model and the phases are as follow:

1. **Requirement gathering and analysis**
2. Implementation
3. Testing
4. Documentation
5. Deployment
6. Maintenance

**Requirement gathering and analysis**

Requirement gathering and analysis is the most important parts of software development and it is usually done by the most skilled and experienced software engineers in the organization. After the requirements are gathered from the client, a scope document is created in which the scope of the project is determined and documented.

Implementation

The software engineers start writing the code according to the client's requirements.

Testing

This is the process of finding defects or bugs in the created software.

Documentation

Every step in the project is documented for future reference and for the improvement of the software in the development process.

Deployment

The software is deployed after it has been approved for release.

Maintenance

Software maintenance is done for future reference. Software improvement and new requirements can take longer than the time needed to create the initial development of the software.

**Waterfall-model**

Waterfall Model was the first process model to be introduced. It is also referred to as a **linear-sequential life cycle model**. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete and in **waterfall model phases** do not overlap.

**Advantages of waterfall model**

* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model as each phase has specific deliverables and a review process.
* Waterfall model works well for smaller projects
* Model phases are processed and completed one at a time as phases do not overlap.

**Disadvantages of waterfall model**

* High amounts of risk and uncertainty.
* Poor model for long and ongoing projects.
* Not a good model for complex and object-oriented projects.
* No working software is produced until late during the life cycle.

**Incremental model**

Incremental model is divided into various builds and multiple development cycles take places here, making the life cycle a “multi-waterfall” cycle. Cycles are then divided up into smaller so as to manage modules easily. Each module will pass through the requirement, design, implementation and lastly the testing phases. A working version of the software is produced during the first module, so we have the working software during the software life cycle. Each subsequent release of the module adds function to the previous release and the process continues until the complete system is achieved.

**Advantages of Incremental mode**

* Generates working software quickly and early during the software life cycle.
* Less costly to change scope and requirements.
* It is easier to test and debug during a smaller iteration.
* Customer can respond to each built.
* Lowers initial delivery cost.

**Disadvantages of Incremental mode**

* Need good planning and design.
* Need a clear and complete definition of the whole system before it can be broken down and built incrementally.

**Agile model**

Agile development model is also a type of [Incremental model](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/). Software is developed in incremental and rapid cycles. Therefore it results in small incremental releases with each release building on previous functionality. Each release is thoroughly [tested](http://istqbexamcertification.com/why-is-testing-necessary/) to ensure that the [software quality](http://istqbexamcertification.com/what-is-software-quality/) is maintained and it is also used for time critical applications. Example of agile development model is the extreme programming (XP)

**Advantages of Agile model**

* Customer satisfaction by rapid, continuous delivery of useful software.
* Customers, developers and testers constantly interact with each other.
* Working software is delivered frequently
* Late changes in requirements can be done

**Disadvantages of Agile model**

* Difficult to assess the effort required at the beginning of the software development life cycle if the software deliverables are larges.
* Lack of emphasis on necessary designing and documentation.
* The project can easily get taken off track if the customer representative is not clear with what is the final outcome that they want.